ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

DIPLOMOVÁ PRÁCA

DÁVID KOŠIČIAR

Decentralizovaná webová aplikácia na online hlasovanie

Vedúci práce: Ing. Tomáš Majer, PhD.

Registračné číslo: 2871/2023

Žilina, 2024

ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

DIPLOMOVÁ

PRÁCA

ŠTUDIJNÝ ODBOR: INFORMATIKA

DÁVID KOŠIČIAR

Decentralizovaná webová aplikácia na online hlasovanie

Žilinská univerzita v Žiline

Fakulta riadenia a informatiky

Katedra matematických metód a operačnej analýzy

Žilina, 2022

ŽILINSKÁ UNIVERZITA V ŽILINE, FAKULTA RIADENIA A INFORMATIKY.

ZADANIE TÉMY DIPLOMOVEJ PRÁCE.

Študijný program: Informačné systémy

Meno a priezvisko
Dávid Košičiar

Osobné číslo
561609

Názov práce v slovenskom aj anglickom jazyku

Decentralizovaná webová aplikácia na online hlasovanie

Decentralized web application for online voting

Zadanie úlohy, ciele, pokyny pre vypracovanie (Ak je málo miesta, použite opačnú stranu)

Cieľ diplomovej práce:
Cieľom práce je navrhnúť a implementovať decentralizovanú webovú aplikáciu na online hlasovanie využívajúcu blockchain

Obsah:
1. Analýza prostriedkov na vývoj decentralizovaných aplikácii v blockchainovej sieti.
2. Analýza potrebných vlastností online hlasovania (bezpečnosť, pseudoanonymita, transparentnosť...) pre rôzne typy hlasovaní (voľby, referendum, petícia, súťaže a pod.).
3. Návrh decentralizovanej aplikácie.
4. Implementácia a overenie aplikácie.

Meno a pracovisko vedúceho DP: Ing. Tomáš Majer, PhD., KMMOA, ŽU
Meno a pracovisko tútora DP:

31 OKT. 2023

garant štud. programu
(dátum a podpis)

Zadanie zaregistrované dňa 31.10.2023 pod číslom 2871/2023 podpis

ČESTNÉ VYHLÁSENIE

Čestne prehlasujem, že som prácu vypracoval samostatne s využitím dostupnej literatúry a vlastných vedomostí. Všetky zdroje použité v diplomovej práci som uviedol v súlade s predpismi.

Súhlasím so zverejnením práce a jej výsledkov.

.....

V Žiline, dňa ..... Meno Priezvisko

Poďakovanie

Chcem sa poďakovať svojmu školiteľovi Ing. Tomášovi Majerovi, PhD. za cenné rady a odbornú pomoc, ktoré mi poskytol pri písaní diplomovej práce.

ABSTRAKT V ŠTÁTNOM JAZYKU

KOŠIČIAR, Dávid: *Decentralizovaná webová aplikácia na online hlasovanie*. [Diplomová práca]. – Žilinská univerzita v Žiline. Fakulta riadenia a informatiky ; Katedra matematických metód a operačnej analýzy. – Vedúci: Ing. Tomáš Majer, PhD. – Stupeň odbornej kvalifikácie: magister. – Mesto: Žilina FRI ŽU, 2024. Počet strán: 74

Diplomová práca sa zameriava na návrh a vývoj decentralizovanej webovej aplikácie pre online hlasovanie. Cieľom práce je definovať výhody a nevýhody tradičného hlasovacieho systému a porovnať ich s online verziou založenou na blockchain technológii. V rámci práce bola vyvinutá webová aplikácia, ktorá pracuje so smart kontraktmi na blockchain. Táto aplikácia umožňuje administrátorovi vytvárať nové hlasovania a pridávať kandidátov. Voľby môžu byť verejné alebo dostupné len overeným voličom. Hlavnou výhodou tejto aplikácie je jej decentralizovaný charakter, čo znamená, že hlasovacie procesy sú transparentné a nezvratné. Okrem toho aplikácia poskytuje možnosť pripojiť vlastné smart kontrakty, čím umožňuje implementáciu rôznych hlasovacích pravidiel a politík. Výsledkom diplomovej práce je plne funkčná webová aplikácia, ktorá môže byť použitá na rôzne typy hlasovaní vrátane verejných volieb, firemných rozhodnutí a mnoho ďalších.

Kľúčové slová: blockchain, online hlasovanie, solidity, web3

ABSTRAKT V CUDZOM JAZYKU

KOŠIČIAR, Dávid: *Decentralized Web Application for Online Voting*. [Master's thesis]. – University of Žilina. Faculty of Management Science and Informatics; Department of Mathematical Methods and Operational Analysis. – Supervisor: Ing. Tomáš Majer, PhD. – Degree of professional qualifications: Master of Science. – City: Žilina, FRI ŽU, 2024. Number of pages: 74

The thesis focuses on the design and development of a decentralized web application for online voting. Its aim is to define the advantages and disadvantages of the traditional voting system and compare them with an online version based on blockchain technology. As part of the work, a web application was developed that operates with smart contracts on the blockchain. This application allows the administrator to create new elections and add candidates. Elections can be either public or available only to verified voters. The main advantage of this application is its decentralized nature, meaning that voting processes are transparent and irreversible. Additionally, the application provides the option to attach custom smart contracts, enabling the implementation of various voting rules and policies. The result of the thesis is a fully functional web application that can be used for various types of voting, including public elections, corporate decisions, and many others.

Key words: blockchain, online voting, solidity, web3

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam skratiek

DApps	Decentralizované aplikácie
ETH	Ethereum
UTXO	Nevyčerpaný transakčný výstup
IPFS	InterPlanetary File System
CID	Content Identifier
DOM	Document Object Model
CI/CD	Kontinuálny proces integrácie a dodania
RPC	Remote Procedure Call

Úvod

Decentralizovaná webová aplikácia na online hlasovanie predstavuje dôležitý krok smerom k modernizácii a demokratizácii volebných procesov. S nástupom blockchain technológie vznikli nové možnosti v oblasti elektronického hlasovania, ktoré ponúkajú transparentnosť, nezvratnosť a bezpečnosť. Táto diplomová práca sa zaoberá návrhom a implementáciou takejto aplikácie, ktorá umožňuje voličom hlasovať online pomocou blockchain technológie.

Mnohí ľudia majú predstavu, že blockchain je relevantný len v oblasti financií, ako sú bankovníctvo alebo kryptomeny. Avšak realita je oveľa zaujímavejšia. Blockchain technológia prináša množstvo nových možností a výhod. Online hlasovanie je len jednou z oblastí, ktorá môže profitovať z tejto technológie a úplne zmeniť spôsob, akým fungujú naše systémy.

Prvú veľkú zmenu priniesla sieť Ethereum, keď umožnila ukladať na svoj blockchain nielen finančné transakcie, ale aj rôzne iné typy dát, ako napríklad smart kontrakty. Tieto kontrakty sú kódy, ktoré je možné vykonať priamo na blockchaine. Jedno z najznámejších použití smart kontraktov je v oblasti Non-Fungible Tokens (NFTs), ktoré výrazne zvýšili povedomie o kryptomenách a ich potenciáli. NFTs umožňujú jedinečné vlastníctvo digitálnych aktív, čo otvára nové možnosti v oblasti digitálneho umenia, virtuálnych svetov a mnoho ďalších.

Online hlasovanie prináša množstvo výhod. Jednou z hlavných úloh blockchainu je zabezpečenie toho, aby dáta nebolo možné úmyselne alebo aj náhodou zmeniť v prospech niekoho iného. Táto vlastnosť zvyšuje dôveryhodnosť k volebnému systému a zabezpečuje, že každý hlas je spravodlivo započítaný. Okrem toho blockchain umožňuje vytvorenie nezvratných záznamov, čo znamená, že hlasovacie údaje sú nezvratné a transparentné. Dáta a všetky transakcie uložené na verejnom blockchaine sú viditeľné, čo umožňuje každému overiť ich integritu a autenticitu.

Pomocou tejto technológie je možné vytvoriť decentralizovaný a transparentný volebný systém, ktorý umožňuje občanom vyjadriť svoj hlas bez obáv o manipuláciu alebo falšovanie výsledkov. Táto práca sa snaží taktiež zvýšiť povedomie o decentralizovaných aplikáciách. Tieto aplikácie bežia na decentralizovaných počítačových sieťach, ako je napríklad už spomínaný blockchain. Oproti tradičným centrálne riadeným aplikáciám,

decentralizované aplikácie majú niekoľko výhod. Prvou výhodou je, že sú odolné voči cenzúre a jednotlivým výpadkom, pretože dáta a funkcie sú distribuované po celej blockchain sieti. Tým sa eliminuje centrálny bod zlyhania.

Cieľom tejto práce je navrhnúť aplikáciu, ktorá bude zabezpečovať základné požiadavky volebných procesov. Táto aplikácia bude postavená na plne decentralizovanej blockchain technológii s využitím Ethereum smart kontraktov. Jej hlavnými funkcionalitami budú vytváranie volieb, pridávanie kandidátov a umožňovanie hlasovania za týchto kandidátov. Tento systém zabezpečí transparentnosť, bezpečnosť a dôveryhodnosť volebných procesov a prispeje k zlepšeniu demokratických postupov.

1 Ciele práce

Cieľom práce je navrhnúť a implementovať decentralizovanú webovú aplikáciu na online hlasovanie využívajúcu blockchain.

1.1 Analýza prostriedkov na vývoj decentralizovaných aplikácii

V tejto časti práce preskúmame dostupné nástroje a technológie na vývoj decentralizovaných aplikácií (DApps) v prostredí blockchainovej siete. Analyzujeme rôzne blockchainové platformy a ich schopnosti podporovať vývoj DApps, ako aj ich výhody a obmedzenia. Na základe tejto analýzy budeme schopní odporučiť konkrétnu blockchainovú platformu alebo kombináciu platformy a nástrojov, ktorá bude najvhodnejšia pre naše ciele pri vývoji decentralizovanej webovej aplikácie na online hlasovanie.

1.2 Analýza potrebných vlastností online hlasovania

Táto časť práce sa bude zaoberať identifikáciou kľúčových vlastností a požiadaviek, ktoré musí spĺňať online hlasovanie, aby bolo dôveryhodné a efektívne. Ako prvý krok, budeme analyzovať existujúce typy hlasovaní a identifikujeme dôvody, prečo sa používajú. Ďalej budeme porovnávať vlastnosti jednotlivých typov hlasovaní a uvedieme ich slabé a silné stránky. Na základe tohto porovnania budeme určovať, ktorý typ systému je vhodný pre konkrétne účely. Ďalej budeme analyzovať aktuálne používané technológie a programy vo svete a skúmať ich fungovanie. Na základe tejto analýzy vyvodíme hlavné vlastnosti, ktoré musí online hlasovanie splniť.

1.3 Návrh decentralizovanej aplikácie

V tejto časti navrhneme architektúru decentralizovanej webovej aplikácie pre online hlasovanie. Popíšeme použité technológie, kľúčové komponenty systému a princípy, na ktorých bude aplikácia fungovať. Na základe tejto architektúry budeme schopní identifikovať a zabezpečiť potrebné prvky pre spoľahlivé a efektívne online hlasovanie.

1.4 Implementácia a overenie aplikácie.

V záverečnej časti práce implementujeme navrhnutú aplikáciu a vykonáme jej overenie. Testovať budeme funkčnosť, bezpečnosť a efektívnosť aplikácie, pričom sa zameriame na dosiahnutie požiadaviek definovaných v analýze. Na základe týchto testov a overení potvrdíme, že aplikácia spĺňa stanovené kritériá a je pripravená na použitie.

2 Súčasný stav

2.1 Fyzické hlasovanie

Fyzické hlasovanie je tradičný spôsob vykonávania volebných práv, pri ktorom volič priamo navštívi volebnú miestnosť. Na začiatku procesu musí volič preukázať svoju identitu pomocou dokladu totožnosti. Po overení údajov a oprávnenia voliča, mu pracovník na volebnom mieste vydá hlasovací lístok. Volič potom fyzicky označí lístok pre zvolených kandidátov a vloží ho do zapečatenej urny. Po skončení volebného dňa sa hlasovacie lístky overia a spočítajú. Následne sa výsledky z volebného miesta odosielajú na ústredné miesto, kde sa vyhodnotia.

Výhody

Výhody fyzického hlasovania zahŕňajú osobnú interakciu s volebným procesom a s pracovníkmi na volebnom mieste, čo môže prispieť k pocitu dôležitosti ich hlasu. Voliči majú väčšiu dôveru vo fyzický volebný proces, keď môžu vidieť odovzdaný hlasovací lístok. Taktiež majú pocit väčšej kontroly nad svojím hlasom a hlasovaním, pretože môžu osobne zabezpečiť, že ich hlas bude odovzdaný správne a bez manipulácií. Ďalšou výhodou je väčšia bezpečnosť, keďže fyzické hlasovanie môže byť menej náchylné k kybernetickým hrozbám alebo manipuláciám v porovnaní s elektronickým hlasovaním. Útoky na hlasovanie sú ťažko škálovateľné. V praxi to znamená, že zvyšovaním počtu falšovaní sa aj zvyšuje náročnosť vykonania tejto činnosti. Zmena jedného volebného lístka je omnoho jednoduchšia ako pri iných typoch volieb, ale ak by útočník chcel zmeniť veľké množstvo hlasov, táto úloha by bola takmer nemožná. Klasické fyzické hlasovanie je tu už po stáročia. Počas tohto obdobia sa už vyskúšali prakticky všetky mysliteľné metódy podvodu. Vďaka tomu sa im už dokážu brániť.

Nevýhody

Náklady na organizáciu fyzických volieb sú často významné. Zahrňujú nájom volebných miestností, výplaty pracovníkov a zabezpečenie materiálov na hlasovanie. Tieto finančné náklady môžu byť pre verejnú správu výzvou, najmä ak je potrebné usporiadať väčšie množstvo hlasovaní ako napríklad referendá. Nízka účasť voličov je ďalšou nevýhodou fyzického hlasovania. Niektorí voliči môžu byť odrádzaní od hlasovania z rôznych dôvodov, ako je zaneprázdnenosť, vzdialenosť volebných miestností alebo obavy z dlhých čakacích radov. Tento nedostatok účasti môže mať za následok nižšiu

reprezentatívnosť volebných výsledkov a znižovať legitimitu výsledkov volieb. Možnosť manipulácie s hlasovacími materiálmi a procesom počítania hlasov je ďalšou významnou nevýhodou fyzického hlasovania. Napriek snahám o zabezpečenie volebných miestností a procesov, stále existuje riziko, že hlasovacie lístky môžu byť počas prepravy alebo volebného dňa manipulované, čo môže ovplyvniť výsledky volieb.

2.2 Elektronické hlasovanie

Elektronické hlasovanie je moderná forma volebného procesu, ktorá využíva elektronické zariadenia na zber a počítanie hlasov. Pri tomto type hlasovania voliči nepoužívajú papierové hlasovacie lístky, ale namiesto toho používajú elektronické hlasovacie terminály umiestnené na volebných miestach. Na týchto termináloch vidia voliči obrazovku s podobným volebným lístkom ako pri tradičnom hlasovaní. Môžu zvoliť svoje voľby pomocou dotykového displeja alebo klávesnice a ich voľby sa automaticky ukladajú do systému. Na konci hlasovania sa potom hlasy automaticky spočítajú.

Výhody

Rýchlosť a efektivita sú jednými z najvýraznejších výhod elektronického hlasovania pomocou terminálov, pretože tento spôsob umožňuje rýchlejšie a jednoduchšie hlasovanie a spracovanie výsledkov. Presnosť a spoľahlivosť patria medzi ďalšie výhody, pretože elektronické systémy minimalizujú možnosť chýb pri počítaní hlasov. Terminály môžu byť prispôsobené rôznym volebným systémom a potrebám rôznych komunít. Dostupnosť elektronického hlasovania zlepšuje účasť voličov a modernizuje volebné procesy. Správne navrhnuté a bezpečnostné systémy elektronického hlasovania môžu byť odolné voči pokusom o manipuláciu alebo podvody. Moderné technológie a bezpečnostné protokoly môžu zabezpečiť integritu a dôvernosť hlasovania.

Nevýhody

Jednou z hlavných nevýhod elektronického hlasovania pomocou terminálov je ich náchylnosť k rôznym formám kybernetických útokov. Tieto útoky môžu ohroziť bezpečnosť hlasovacieho procesu a integritu volebných výsledkov. Nájdenie jednej chyby v systéme môže ohroziť celý hlasovací proces. Taktiež ovplyvnenie celého výsledku je rovnako náročné ako zmena jedného hlasu. Útočník sa nemusí ani nachádzať v rovnakej krajine ako sa odohrávajú voľby a je takmer nemožne identifikovať páchateľa. Existuje viacero problémov ktoré je náročné vyriešiť a zabezpečiť.

Audit softvéru a hardvéru je prvým problémom. Aj keď môžeme mať open-source softvér, ktorý bol dôkladne overený a považovaný za bezpečný, problém vzniká pri zabezpečení, že tento softvér je skutočne nainštalovaný na všetkých hlasovacích termináloch a nebol upravený. Ak by aj bol softvér spoľahlivý, stále môže byť pochybnosť o tom, či bol skutočne nainštalovaný na všetkých zariadeniach a či nebol počas používania zmenený. Riešením tohto problému by mohol byť vývoj programu, ktorý by overoval prítomnosť a nezmenený stav softvéru, ale to by len presunulo problém inde.

Ďalším problémom je otázka bezpečnosti terminálov počas hlasovania. Na to, aby bolo zabezpečené, že hlas voliča zostane anonymný, je potrebné, aby bolo zariadenie používané iba voličom a nikým iným. To znamená, že musí byť umiestnený v izolovanej miestnosti, aby sa zabránilo prístupu iných osôb, ktoré by mohli ovplyvniť hlasovanie. Okrem toho je potrebné zabezpečiť, aby nikto nemohol zmeniť žiadne parametre na termináli počas hlasovania, čo by mohlo ohroziť jeho integritu a spoľahlivosť. Tieto opatrenia sú nevyhnutné na zabezpečenie bezpečného a spoľahlivého hlasovacieho procesu.

Existujú tri hlavné spôsoby, ako presunúť hlasovacie dáta z volebných miestností na centrálnu prevádzku, kde budú spočítané. Jedným zo spôsobov je zapečatiť celé hlasovacie zariadenie a prepraviť ho vozidlom na stanovené miesto. Tento prístup si vyžaduje veľké množstvo práce a logistiky. Druhou možnosťou je uložiť hlasovacie dáta na prenosné úložisko a prepraviť ho na finálne miesto. Avšak tento spôsob sa obvykle nepoužíva kvôli bezpečnostným obavám a časovej náročnosti spojenej s prenosom úložiska. Najpoužívanejším riešením je odoslanie výsledkov hlasovania z hlasovacích terminálov cez internet. Tento postup však prináša obrovské bezpečnostné riziká, ktoré musia byť primerane zvládnuté a zabezpečené.

2.3 Online/Internetové hlasovanie

Zatiaľ len malé množstvo krajín začalo používať internetové hlasovanie. V tomto type hlasovania môže byť použité každé zariadenie s pripojením na internet.

V roku 2004 Výbor ministrov Rady Európy schválil odporúčanie o normách pre elektronické voľby a dňa 14. júna 2017 prijal novú verziu tohto odporúčania. Odporúčanie sa týka volebných strojov, skenerov hlasovacích lístkov, ako aj online volebných systémov. Spolu s priloženými pokynmi pre uplatnenie noriem vysvetľuje podmienky, ktoré musia systémy pre e-hlasovanie splniť, aby sa zabezpečilo dodržiavanie základných princípov

volieb. Tento rámec je vynikajúcim východiskom pre všetky krajiny, ktoré plánujú zaviesť e-hlasovanie. [1].

Z tohto usmernenia sa dajú vyvodiť základné vlastnosti ktoré by mali byť splnené pri návrhu a implementácii vlastného systému. Tieto vlastnosti zahŕňajú bezpečnosť, dôveryhodnosť, transparentnosť, pseudo-anonymitu, prístupnosť a jednoduchosť použitia. Bezpečnosť zaručuje, že hlasovacie údaje zostanú chránené pred neoprávneným prístupom alebo manipuláciou. Dôveryhodnosť znamená, že systém zabezpečuje správne a presné zaznamenanie a počítanie hlasov. Transparentnosť je dôležitá pre verejnú dôveru a znamená, že celý proces hlasovania je viditeľný a verifikovateľný. Pseudo-anonymita zabezpečuje, že hlas voliča ostane anonymný, aby sa ochránila jeho súkromie. Prístupnosť znamená, že systém je dostupný pre všetkých oprávnených voličov bez ohľadu na ich fyzické alebo technické schopnosti. Jednoduchosť použitia zaručuje, že hlasovanie je intuitívne a ľahko pochopiteľné pre všetkých používateľov.

2.3.1 Online hlasovanie v Estónsku

Estónsko bolo prvou krajinou na svete, ktorá zaviedla internetové hlasovanie do svojich všeobecných volieb. Elektronické hlasovanie s viazanými výsledkami sa v Estónsku praktizuje od roku 2005. Toto hlasovanie je populárne predovšetkým pre svoju efektivitu a pohodlnosť. Dnes sa približne tretina všetkých hlasov odošle prostredníctvom internetu. Systém, ktorý sa v súčasnosti používa, bol dokončený pred miestnymi voľbami v roku 2017 a bol vyvinutý podľa nového rámca pre elektronické hlasovanie. [2]

Priebeh

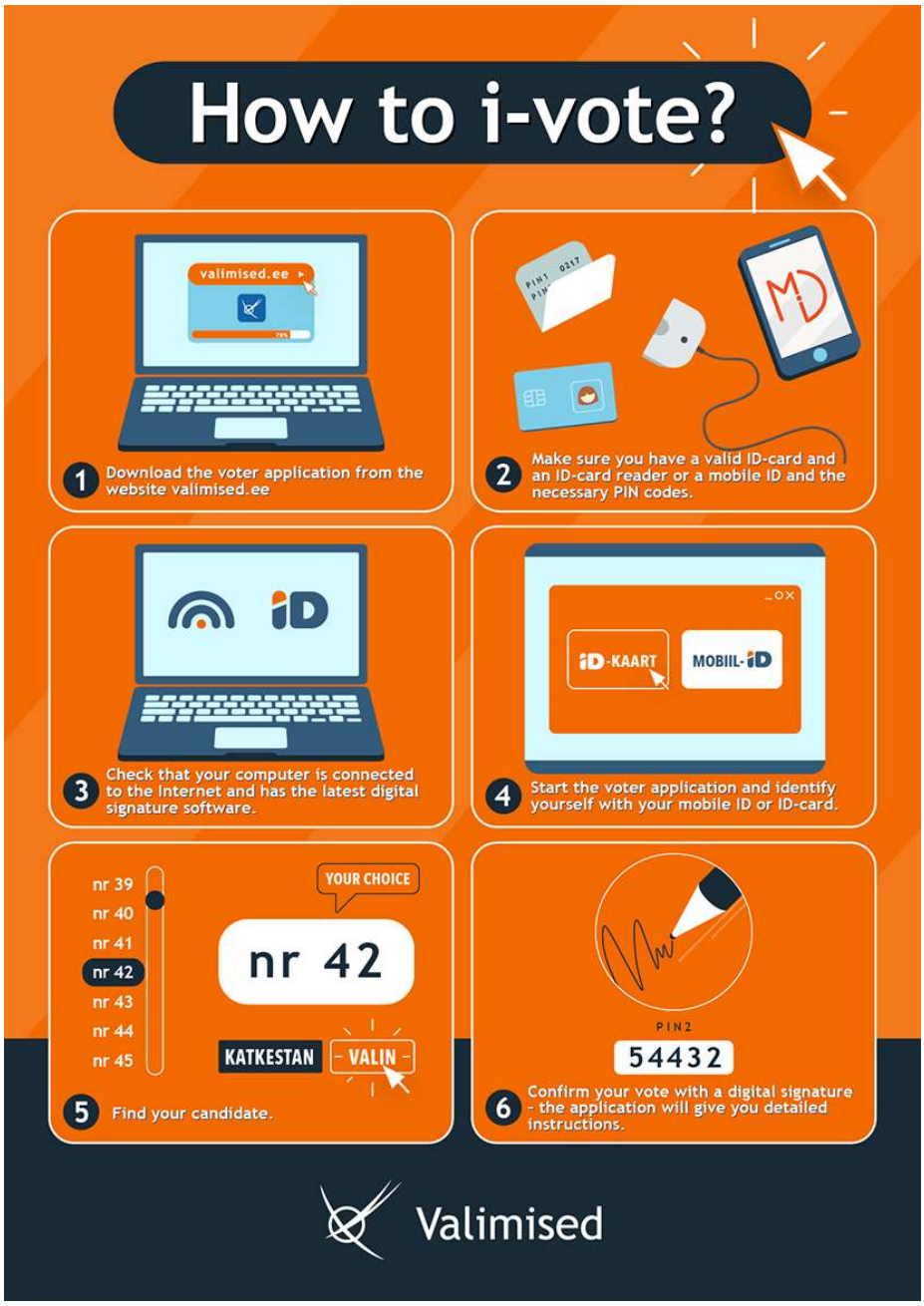
Pred začiatkom hlasovania príslušný úrad pripraví systém I-voting a zverejní aplikáciu pre voličov potrebnú na hlasovanie na webovej stránke "valimised.ee". Elektronické hlasovanie začína v pondelok volebného týždňa o 9:00 hod. a končí v sobotu večer o 20:00. Na hlasovanie je potrebný počítač pripojený k internetu a občiansky preukaz, mobilný občiansky preukaz alebo digitálny identifikačný dokument s platnými certifikátmi a PIN kódmi.

Ďalej je potrebné stiahnuť aplikáciu pre voličov na počítač. Aplikácia automaticky overí oprávnenosť voliča a zobrazí mu správny zoznam kandidátov. Po výbere voliča aplikácia zašifruje jeho hlas. Volič potvrdí svoje hlasovanie digitálnym podpisom a aplikácia pre voliča prepošle hlas na server pre zhromažďovanie hlasov. Súčasne nezávislá registračná

služba každý hlas ošetrí časovou pečiatkou, ktorá umožňuje neskôr overiť, že všetky hlasy boli skutočne odoslané na centrálny server. [2]

Overenie

Volič si môže overiť, či jeho hlas bol správne odoslaný a prijatý, pomocou samostatnej aplikácie na inteligentnom zariadení. Hlasy sú zašifrované pomocou vhodného a aktuálneho kryptografického algoritmu. Presná špecifikácia algoritmu je určená Štátnym volebným úradom pred každými voľbami. Hlas je zašifrovaný pomocou dvoch šifrovacích kľúčov. Aplikácia pre voliča používa verejný šifrovací kľúč hlasu. K otvoreniu hlasu je potrebný kľúč na otvorenie hlasu; prístup k nemu majú len členovia Národného volebného výboru. [2]



Obr. 1 Návod na hlasovanie v Estónsku

2.3.2 Švajčiarsko

Momentálne sa uskutočňuje testovanie internetového hlasovania v niektorých kantónoch Švajčiarska. Hlasy odovzdané prostredníctvom tohto hlasovania sa pripočítavajú k celkovému výsledku. Rozsah skúšok je ešte stále obmedzený. Maximálne 30 percent oprávnených voličov má povolené používať elektronické hlasovanie v akomkoľvek konkrétnom kantóne, pričom celkový štátny limit je stanovený na 10 percent. Súčasné čísla hlasovania stále výrazne zaostávajú za týmito kvótami. Z internetového hlasovania budú mať prospech hlavne voliči s postihnutím. Napríklad občanom so zrakovými problémami

umožní jednoduchšie hlasovať bez asistencie. Možnosť hlasovať online tiež oslobodí Švajčiarov žijúcich v zahraničí od závislosti na miestnych poštových službách. [3]

Priebeh

Ak ste súčasťou skúšobného elektronického hlasovania, obdržíte volebný preukaz s vašimi osobnými overovacími kódmi. Tieto kódy sa použijú na odovzdanie hlasu a následne overenie, že bol správne odoslaný. Volebný preukaz obsahuje pokyny, ako pristupovať k elektronickému portálu kantónu, ako aj ako odovzdať a overiť hlas. Dostanete taktiež informácie o lehote pre elektronické hlasovanie. Hlas bude zašifrovaný. Úrady nebudú môcť sledovať, ako konkrétny občania hlasovali. [3]

2.3.3 Voľby do obecných samospráv v Ontáriu

Internetové hlasovanie je široko používané pri obecných voľbách v provinciách Ontária a Nového Škótska. Voľba spôsobu hlasovania je na miestnych úradoch. V Novom Škótsku v roku 2016 používalo internetové hlasovanie 20 miestnych úradov. Na obecných voľbách v Ontáriu v roku 2018 použilo internetové hlasovanie 194 z 444 miestnych úradov. Presun k e-hlasovaniu je hlavne z dôvodu prístupnosti a pohodlia. Taktiež sa zvýšil počet dní na hlasovanie. Ďalším veľkým dôvodom je zníženie počtu potrebných zamestnancov v porovnaní s klasickým hlasovaním. Voliči si taktiež chvália tento spôsob. V prieskume vykonanom na 33 tisíc občanoch, 95 percent bolo spokojných s procesom ktorý volili online. Zatiaľ čo iba 68 percent bolo spokojných pri klasickom fyzickom hlasovaní [4]. Niektorí odborníci ale tvrdia, že e-hlasovanie má aj svoje riziká a nevýhody. Majú obavy z rýchlosti, akou jurisdikcie prijímajú nové technológie ako jediný spôsob hlasovania. Každé voľby sa nájdu taký, čo sa snažia manipulovať systém a nie vždy sú to iba hackeri. Očakáva sa, že Kanada taktiež zavedie jednotné dobrovoľné normy a smernice pre e-hlasovanie, ako to urobila Európa. Kanadská vláda taktiež potvrdila, že v roku 2017 nemá žiadne plány na zavedenie internetového hlasovania pri národných voľbách, a to platí aj v roku 2024. [5]

Problémy a obavy

Aleksander Essex, profesor na Western University, ktorý sa špecializuje na kybernetickú bezpečnosť, vyzýva mestské samosprávy v Ontáriu, aby skutočne pochopili, ako funguje internetové hlasovanie, predtým ako bude implementované. Varuje, že technológia internetového hlasovania má malú transparentnosť a neexistuje žiadna záruka, že hlasy sú správne započítané. Problém vyplýva z faktu, že niektoré mestské samosprávy ani nevedia, kde sa nachádzajú servery, ktoré používajú na hlasovanie, dokonca sa server nemusí ani nachádzať v Kanade. [6]

Neexistuje žiadny komplexný zoznam častí používajúcich internetové hlasovanie. Nemá definované normy, obmedzenú transparentnosť a žiadnu kontrolu nad tým, či sú hlasy započítané správne. Okrem toho zanecháva voľby náchylné na hackovanie prostredníctvom malvéru, najmä na domácom počítači. Niektoré časti v Ontáriu už preukázali, že sú náchylné na kybernetické útoky. Niektoré provincie internetové hlasovanie úplne zavrhli. [6]

Dean Smith, prezident spoločnosti Intelivote, Dartmouth NS, ktorá bola najatá na riadenie internetového hlasovania v provinciách Ontária, tvrdí, že jeho spoločnosť duplikuje údaje na ďalšom serveri, aby sa zabránilo problémom so zariadeniami. Profesor Essex zdôrazňuje, že napriek ubezpečeniam dodávateľov neexistujú žiadne záruky, že systémy na internetové hlasovanie skutočne odzrkadľujú zámery voličov. Taktiež tvrdí, že napriek preferencii hlasovať online je ručné počítanie papierových hlasov najlepším riešením. [6]

2.3.4 Blockchain hlasovanie

Blockchain hlasovanie je inovatívna metóda zaznamenávania hlasov voličov pomocou technológie blockchain. Je to distribuovaný systém, ktorý umožňuje zaznamenávať a overovať transakcie alebo záznamy v decentralizovanej sieti počítačov. Pri hlasovaní sa každý hlas zaznamenáva ako digitálny záznam, nazývaný blok, ktorý je spojený s ostatnými blokmi v reťazi. Tieto bloky sú následne overované a uložené na množstve počítačov v sieti, čím sa zaisťuje ich integrita a transparentnosť.

Hlavnou výhodou je jeho bezpečnosť a nemennosť. Keďže dáta sú uložené na viacerých miestach v sieti, je takmer nemožné ich upraviť alebo odstrániť. Každý blok v blok obsahuje digitálny podpis. Tento podpis zabezpečuje autentickosť každého bloku. Toto robí hlasovanie veľmi odolným voči podvodom a manipuláciám. Okrem toho je blockchain hlasovanie transparentné a dôveryhodné. Každý účastník má možnosť sledovať a overiť svoj hlas prostredníctvom verejne dostupných blockchain záznamov. Tým sa zvyšuje dôvera vo volebný proces a znižuje sa riziko sporov ohľadom volebných výsledkov.

Ďalšou výhodou je efektívnosť a rýchlosť. Vďaka automatizovanému zaznamenávaniu a overovaniu hlasov sa čas potrebný na vyhodnotenie volebných výsledkov výrazne skracuje. V neposlednom rade je toto hlasovanie aj odolné voči vonkajším útokom a kybernetickým hrozbám. Vzhľadom na distribuovanú povahu blockchainu je obťažné narušiť integritu systému.

3 Použité technológie

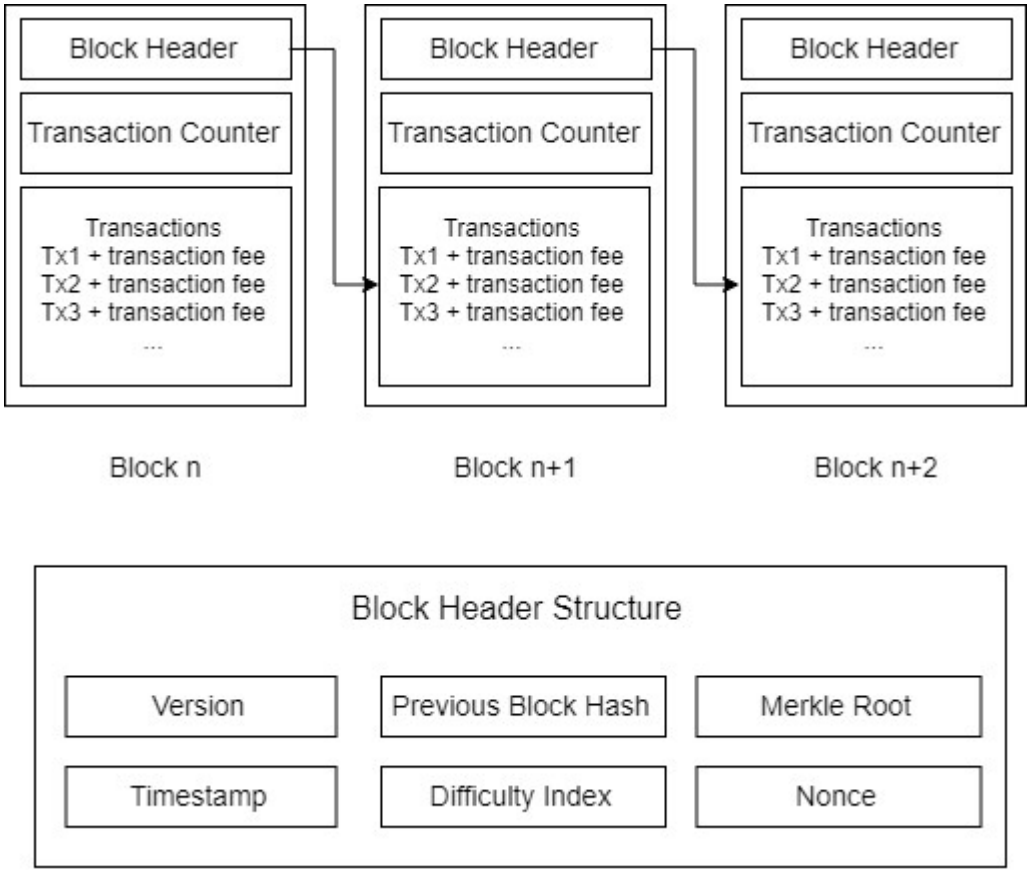
3.1 Bitcoin/Ethereum

Na určenie vhodnej technológie je nevyhnutné pochopiť základné princípy fungovania Bitcoinu. Bitcoin, ako prvá kryptomena, využíva blockchain, decentralizovaný systém záznamu transakcií. Ethereum, na druhej strane, vychádza z konceptu Bitcoinu a rozširuje ho o možnosť vykonávania kódu. Tento kód môže byť implementovaný ako backendová služba pre decentralizovanú aplikáciu. Ethereum poskytuje programovateľný blockchain, kde kontrakty môžu spolupracovať s akýmikoľvek komplexnými funkciami. Tieto kontrakty, známe ako smart kontrakty, sú implementované v programovacom jazyku Solidity a sú schopné definovať rôzne procesy a podmienky vykonávania. Tento mechanizmus umožňuje vytváranie decentralizovaných aplikácií, ktoré sú odolné voči manipulácii a cenzúre, čo je kľúčovým aspektom pri online hlasovaní.

3.1.1 Bitcoin

Obchodovanie na internete spoľahlivo funguje vďaka úlohe finančných inštitúcií ako dôveryhodných sprostredkovateľov elektronických platieb. Napriek tomu tento systém trpí nedostatkami, keďže neumožňuje úplne nezvratné transakcie, a to preto, že finančné inštitúcie nemôžu zabrániť riešeniu sporov. Bitcoin predstavuje riešenie týchto problémov, ponúkajúc elektronický platobný systém založený na kryptografickom dôkaze namiesto dôvery. Jeho transakcie sú chránené náročnými výpočtami, ktoré zabraňujú praktickému zmeneniu alebo odstráneniu záznamu z transakčného bloku. Dôležitou podmienkou je, aby aspoň polovica výpočtových kapacít patrila poctivým uzlom a nie útočníkom. [7]

Blockchain



Obr. 2 Bloková štruktúra Bitcoinu

. Na obrázku je zobrazená štruktúra blokov obsahujúcich informácie o jednotlivých transakciách. Každý blok obsahuje odkaz na predchádzajúci blok vo forme hashu. Ak sa zmení aj jediná informácia v bloku, zmení sa aj jeho hash. Táto vlastnosť zabezpečuje, že ak by sme chceli upraviť dáta v jednom bloku, museli by sme upraviť aj všetky nasledujúce bloky. Každý uzol v sieti uchováva kópiu blockchainu, čím sa zaisťuje jeho integrita a nezvratnosť.

Transakcia

Transakcia v Bitcoinovej sieti predstavuje prenos hodnoty medzi Bitcoinovými peňaženkami, ktorý je zapísaný do blockchainu. Peňaženky uchovávajú tajný údaj, známy ako privátny kľúč alebo seed, ktorý sa používa na podpisovanie transakcií a matematicky dokazuje ich pôvod od vlastníka peňaženky. Tento podpis zároveň zabraňuje akýmkoľvek zmenám v transakcii po jej vystavení. Všetky transakcie sú následne odosielané do siete a obvykle sú potvrdené v priebehu 10-20 minút, pomocou procesu známeho ako ťaženie. [8]

Ťaženie

Ťaženie je distribuovaný systém, ktorý sa používa na potvrdenie čakajúcich transakcií a ich zahrnutie do blockchainu. Zabezpečuje chronologické usporiadanie, chráni neutralitu siete a umožňuje dohodu rôznym počítačom o stave systému. Transakcie musia byť zahrnuté do bloku, ktorý spĺňa veľmi prísne kryptografické pravidlá, ktoré sú overené sieťou. Tieto pravidlá bránia modifikácii predchádzajúcich blokov, čo by zneplatnilo všetky nasledujúce bloky. Ťažba tiež vytvára ekvivalent súťažného lotériového systému, ktorý bráni jednotlivcom v jednoduchom pridávaní nových blokov po sebe do blockchainu. Týmto spôsobom žiadna skupina ani jednotlivci nemôžu ovplyvniť obsah blockchainu alebo nahrať časti blockchainu, aby zrušili svoje vlastné transakcie.. [8]

Limity skriptovania v Bitcoine

Bitcoin je napísaný v takzvanom "Turing incomplete" jazyku, čo znamená, že rozumie len malej sade príkazov ako napríklad kto poslal koľko peňazí komu. Hlavná kategória príkazov, ktoré chýbajú sú "loopy". Toto je vykonané s cieľom zabrániť nekonečným slučkám počas overovania transakcií. Teoreticky je to možné obísť, ale prakticky to vyžaduje obrovské množstvo pamäte. [9]

UTXO predstavuje určité množstvo kryptomeny, ktoré bolo autorizované odosielateľom a ktoré môže príjemca minúť. V praxi to znamená, že vlastníkom kryptomeny môže byť nielen verejný kľúč, ale aj skript. Neexistuje ale spôsob, ako poskytnúť detailnú kontrolu nad množstvom UTXO, ktoré je možno vybrať zo skriptu. [9]

UTXO môžu byť buď minuté alebo neminuté. Neexistuje príležitosť pre viacstupňové kontrakty alebo skripty, ktoré udržiavajú akýkoľvek iný vnútorný stav. Toto neumožňuje vytvárať viacstupňové možnosti kontraktov, ponuky decentralizovaných výmenných kurzov alebo dvojstupňové protokoly kryptografických záväzkov (potrebné pre bezpečné výpočtové odmeny). Taktiež to znamená, že UTXO môžu byť použité iba na vytvorenie jednoduchých, jednorazových kontraktov. [9]

UTXO nevidia blockchainové údaje, ako je nonce, časová pečiatka a hash predchádzajúceho bloku. Toto vážne obmedzuje aplikácie tým, že skriptovaciemu jazyku odopiera potenciálne cenný zdroj náhodnosti. [9]

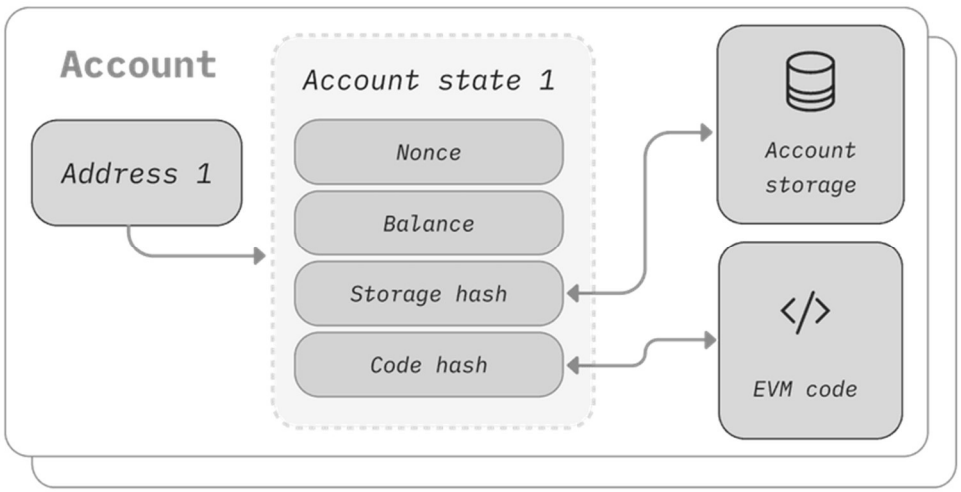
3.1.2 Ethereum

Cieľom Etherea je vytvoriť alternatívny protokol na vývoj decentralizovaných aplikácií, poskytujúci odlišný súbor výhod, ktoré sa považujú za veľmi užitočné pre veľkú skupinu decentralizovaných aplikácií, s osobitným dôrazom na situácie, kde je dôležitý rýchly vývoj, bezpečnosť a schopnosť rôznych aplikácií veľmi efektívne interagovať. Ethereum to dosahuje vytvorením takzvanej ultimatívnej abstraktnej základnej vrstvy: blockchain s integrovaným "Turing-complete" programovacím jazykom, ktorý umožňuje vytvárať smart kontrakty a decentralizované aplikácie. V nich je možné vytvoriť vlastné arbitrárne pravidlá pre vlastníctvo, formáty transakcií a funkcie prechodu stavov. Smart kontrakty obsahujú hodnotu a uvoľnia ju iba v prípade, že sú splnené určité podmienky. Môžu byť taktiež postavené na platforme s oveľa väčšími právomocami ako to ponúka Bitcoin vďaka pridaným vlastnostiam "Turing-complete" jazyka. [9]

Ethereum Accounts

V Ethereu je stav tvorený objektmi nazývanými "accounts", pričom každý účet má adresu s dĺžkou 20 bajtov a prechodmi stavu, ktoré sú priamymi presunmi hodnôt a informácií medzi účtami. [9] Účet v Ethereu obsahuje štyri položky:

- **Nonce**
- **Eth balance**
- **Code**
- **Storage**



Obr. 3 Ethereum Account

"Ether" je hlavná interná mena v Ethereu a slúži na úhradu poplatkov za transakcie. Všeobecne existujú dva typy účtov: **externé účty**, ktoré sú ovládané privátnymi kľúčmi, a **kontraktové účty**, ktoré sú ovládané svojím kódom. Externý účet nemá kód a správy je možné posielať z externého účtu vytvorením a podpísaním transakcie. V prípade kontraktového účtu sa pri prijatí správy aktivuje jeho kód, čo mu umožňuje čítať a zapisovať do vnútorného úložiska a zaslať ďalšie správy alebo vytvoriť kontrakty ďalej. [9]

3.1.3 Decentralizované aplikácie

A decentralizovaná aplikácia (DApp) je aplikácia postavená na decentralizovanej sieti, ktorá kombinuje smart kontrakt a front end užívateľské rozhranie. Na platforme Ethereum sú smart kontrakty prístupné a transparentné - podobne ako open APIs - takže DApp môže dokonca obsahovať smart kontrakty, ktoré napísal niekto iný. [10]

Tieto aplikácie majú svoj backend kód bežiaci na decentralizovanej peer-to-peer sieti na rozdiel od klasických aplikácií, kde backend kód funguje na centralizovaných serveroch. Decentralizovaná aplikácia môže mať front end kód a užívateľské rozhrania napísané v ľubovoľnom jazyku. Okrem toho, jeho front end môže byť hostovaný na decentralizovanom úložisku ako napríklad IPFS. [10] Decentralizované aplikácie sú :

- **Decentralizované** - DApps fungujú na platforme Ethereum, otvorenej verejnej decentralizovanej platforme, kde žiadna osoba alebo skupina nemá kontrolu [10]
- **Deterministické** - DApps vykonávajú rovnakú funkciu bez ohľadu na prostredie, v ktorom sú vykonávané [10]
- **Turing kompletné** - DApps môžu vykonávať ľubovoľnú akciu za predpokladu, že majú potrebné zdroje [10]
- **Izolované** - DApps sú vykonávané v virtuálnom prostredí známom ako Ethereum Virtual Machine, takže ak smart kontrakt obsahuje chybu, neovplyvní to normálne fungovanie blockchainovej siete. [10]

Výhody

- **Zero downtime** – Po nasadení smart kontraktu na blockchain bude sieť ako celok vždy schopná obslúžiť klientov hľadajúcich interakciu s kontraktom. Hackeri preto nemôžu vykonávať takzvané "denial-of-service" útoky mierené na individuálne aplikácie. [10]

- **Súkromie** – Na nasadenie alebo interakciu s decentralizovanou aplikáciu nepotrebujete poskytnúť skutočnú identitu. Prihlasovanie a interagovanie funguje pomocou Ethereum účtov, ktoré sú sprostredkované externými aplikáciami nazývanými peňaženky. [10]
- **Odolnosť voči cenzúre** – Žiadna entita v sieti nemôže blokovať používateľov pri zasielaní transakcií, nasadzovaní kontraktov alebo čítaní údajov z blockchainu. [10]
- **Úplná integrita údajov** – Údaje uložené na blockchainu sú nezmeniteľné a neodvolateľné vďaka kryptografickým funkciám. Kybernetický útočník nemôžu falšovať transakcie ani iné údaje, ktoré už boli zverejnené. [10]
- **Dôveryhodné výpočty/overiteľné správanie** – Smart kontrakty môžu byť analyzované a sú garantované, že sa budú vykonávať v predvídateľných spôsoboch, bez potreby dôvery voči centrálnej autorite. To neplatí v tradičných modeloch; napríklad pri používaní internetových bankových systémov musíme dôverovať, že finančné inštitúcie nezneužijú naše finančné údaje, nezasahujú do záznamov alebo nebudú hacknuté. [10]

Nevýhody

- **Údržba** - DApps môžu byť ťažko udržiavateľné kvôli obťažnosti modifikácie kódu a dát po ich nasadení na blockchain. Pre vývojárov je náročné meniť a zlepšovať kód aplikácie alebo vnútorné parametre. Tento problém je zvýraznený hlavne keď sa objavia chyby alebo bezpečnostné problémy v predchádzajúcich verziách. [10]
- **Škálovateľnosť** – Je náročné dosiahnuť škálovateľnosť týchto typov aplikácii. Ethereum sa snaží o bezpečnosť, integritu, transparentnosť a spoľahlivosť. To vyžaduje, aby každý uzol spustil a uchoval každú transakciu. Okrem toho, konsenzus typu proof-of-stake tiež zaberie čas. [10]
- **Preťaženie siete** - Nadmerné používanie výpočtových zdrojov jednou aplikáciu môže spôsobiť preťaženie a následné spomalenie celej siete. V súčasnosti sieť dokáže spracovať len približne 10-15 transakcií za sekundu, čo vedie k hromadeniu neoverených transakcií, ak rýchlosť transakcií presahuje túto kapacitu. [10]

- **Používateľská skúsenosť** - Návrh používateľsky prívetivých systémov môže byť náročné. Pre priemerného používateľa sa môže zdať, že náročné a zložité nastaviť potrebné nástroje na bezpečnú interakciu s blockchainom. [10]
- **Centralizácia** -Používateľsky a vývojársky priateľské riešenia postavené na základnej vrstve Etherea sa nakoniec môžu javiť ako centralizované služby. Napríklad takéto služby môžu ukladať kľúče alebo iné citlivé informácie na serverovej strane, poskytovať front end pomocou centralizovaného servera alebo vykonávať dôležitú obchodnú logiku na centralizovanom serveri pred zapísaním do blockchainu. Centralizácia eliminuje mnohé (ak nie všetky) výhody blockchainu oproti tradičnému modelu.. [10]

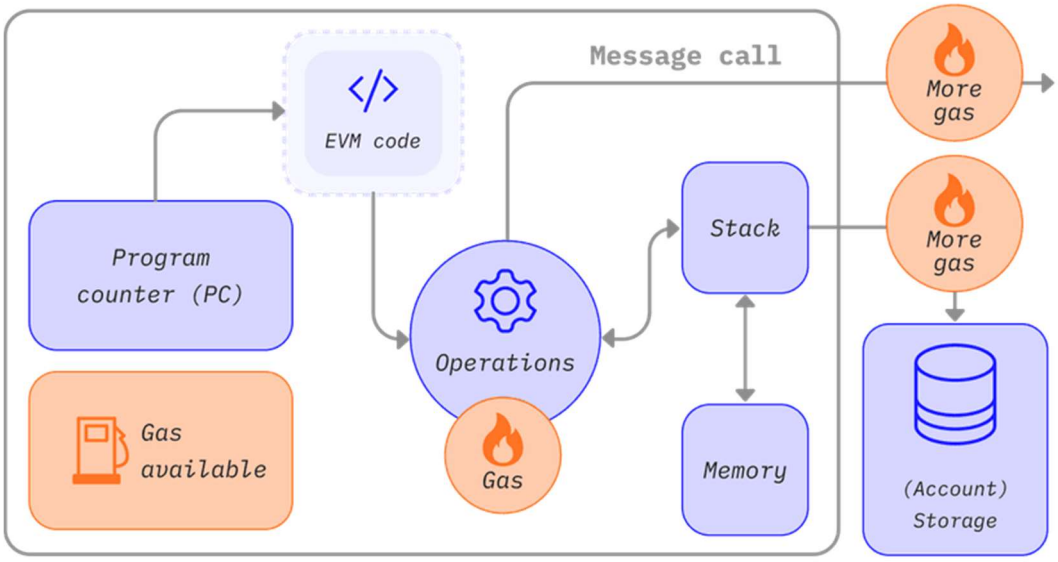
Tab. 1 Rozdiel medzi centralizovaným a decentralizovaným systémom

Centralizované systémy	**Decentralizované systémy**
Informácie sa rýchlo šíria, keďže šírenie je riadené centrálnou autoritou s veľkými výpočtovými zdrojmi.	Najvzdialenejší účastníci v sieti môžu byť vzdialení mnoho uzlov od seba. Informácie poslané z jednej strany siete môžu dlho trvať, kým dosiahnu druhú stranu.
Vyšší výkon	Nižší výkon
Konflikty rieši centrálny server	Potrebný protokol na riešenie konfliktov
Jednotný bod zlyhania	Sieť funguje aj po výpadku väčšieho množstva uzlov
Jednoduchá koordinácia	Zložitá koordinácia
Centrálny server môže cenzurovať dáta	Takmer nemožná cenzúra
Centrálny server rozhoduje kto sa môže zúčastniť	Každý sa môže zúčastniť

3.1.4 Gas

Gas sa odkazuje na jednotku, ktorá meria množstvo výpočtového úsilia potrebného na vykonanie konkrétnych operácií v sieti Ethereum. Keďže každá transakcia na Ethereu vyžaduje výpočtové zdroje na jej vykonanie, tieto zdroje musia byť zaplatené, aby sa zabránilo spamu v sieti Ethereum a aby sa zabránilo zacykleniu v nekonečných výpočtových

slučkách. Platba za výpočet sa uskutočňuje vo forme poplatku za gas. Tento poplatok je množstvo gas-u použité na vykonanie určitej operácie, násobené nákladmi na jednotku gas-u. Poplatok sa platí bez ohľadu na to, či transakcia uspeje alebo zlyhá. [11]



Obr. 4 Gas

Poplatky za gas musia byť zaplatené v domácej mene Etherea, Ether (ETH). Ceny sú zvyčajne udávané v gwei, ktorý je jednotkou denominácie ETH. Každý gwei je rovný jednej miliardtine etheru (0,000000001 ETH alebo 10-9 ETH). Napríklad namiesto toho, aby bolo povedané, že poplatky stoja 0,000000001 etheru, je možné povedať, že gas stojí 1 gwei. Slovo „gwei“ je skrátenina slova „giga-wei“, čo znamená „miliarda wei“. Jedno gwei je rovné jednej miliarde wei. Wei (pomenovaná podľa Weia Daiho, tvorcu b-money) je najmenšia jednotka ETH. [11]

Výpočet poplatkov za gas

Možné je nastaviť ľubovoľné množstvo gasu, ktoré je žiadateľ ochotný zaplatiť pri odosielaní transakcie. Ponúkaním určitého množstva gasu sa uchádza o zaradenie svojej transakcie do ďalšieho bloku. Ak je ponúknuté množstvo príliš malé, validátori sú menej ochotný vybrať transakciu na zaradenie, čo znamená, že transakcia sa môže vykonať neskoršie alebo vôbec nie. Ak je ponúknuté príliš veľa, zbytočne sa premrháva ETH. [11]

Z tohto vyplýva, že je potrebné správne odhadnúť množstvo. Gas sa delí na dva komponenty: základný a prioritný poplatok. Základný poplatok určuje protokol. Je potrebné ponúknuť aspoň toto množstvo aby sa transakcia mohla považovať za validnú. Prioritný príplatok je niečo ako prepitné ktoré sa snaží nalákať validátorov na rýchlejšie spracovanie.

Odporúčané množstvo prioritného poplatku sa taktiež mení aj na základe vyťaženia siete. [11] Bežný užívateľ ale nemusí riešiť žiadny výpočet množstva gasu na transakciu ktorú chce vykonať. Takmer každá používaná aplikácia na správu Ethereum účtu odhaduje cenu transakcie za svojho používateľa. Tu môže užívateľ nastaviť maximálnu hodnotu, ktorú je ochotný zaplatiť. Aj keď transakcia zahŕňa limit, akýkoľvek nepoužitý gas v transakcii je vrátený používateľovi.

Prečo existujú poplatky za gas

V skratke, poplatky za gas pomáhajú udržiavať sieť Ethereum bezpečnú. Vyžadovaním poplatku za každý výpočet vykonaný v sieti sa bráni zlým aktérom v spamovaní siete. Aby sa vyhlo náhodným alebo útočným nekonečným slučkám alebo inému zbytočnému výpočtovému plytvaniu v kóde, je každá transakcia povinná stanoviť limit, koľko výpočtových krokov sa môže vykonať. [11] Preto je veľmi dôležité písať smart kontrakty takým spôsobom, aby boli čo najviac optimalizované v počte krokov a množstva pamäte, ktoré používajú.

3.1.5 Ethereum siete

Siete Etherea sú skupiny spojených počítačov, ktoré komunikujú pomocou protokolu Ethereum. Existuje iba jedna sieť Ethereum Mainnet. Je tu ale množstvo nezávislých sietí, ktoré dodržiavajú rovnaké pravidlá tohto protokolu. Sú vytvárané hlavne na testovacie účely. Existuje mnoho nezávislých "sietí", ktoré dodržiavajú protokol, ale neinteragujú medzi sebou. Dokonca je ich možné spustiť aj lokálne na svojom počítači na testovanie smart kontraktov a aplikácií web3.

Jeden Ethereum účet funguje naprieč rôznymi sieťami, ale finančný zostatok účtu a jeho história transakcií je pre každú sieť vlastná. Na testovacie účely je užitočné vedieť, ktoré siete sú k dispozícii a ako získať testovací ETH na skúšanie transakcií a smart kontraktov. Všeobecne ale platí, že z bezpečnostných dôvodov sa neodporúča opätovné používanie účtov z hlavnej siete na testovacích sieťach alebo naopak.

Testnets

Každý kontrakt by mal byť najskôr otestovaný a overený na testovacích siatiach pred nasadením na mainnet. Je to z dôvodu, že kód už nasadeného kontraktu nie je možné zmeniť. ETH na testovacích sieťach nemá reálnu hodnotu. Získať ho je možné zadarmo pomocou služby známej ako faucet. Tu sa zadá iba adresa účtu a na ten sa pripíše malé množstvo testovacej kryptomeny.

Dve verejné testovacie siete, ktoré sú aktuálne udržiavané vývojármi, sú Sepolia a Goerli. Sepolia je sieť pre vývojárov kontraktov na testovanie ich aplikácií. Sieť Goerli umožňuje vývojárom protokolov testovať aktualizácie siete a validátorom testovať prevádzku validátorov.

Sepolia

Sieť Sepolia používa povolený súbor validátorov. Je pomerne nová, čo znamená, že jej stav a história sú veľmi malé. To znamená, že sieť sa rýchlo synchronizuje a pre prevádzkovanie uzla na nej je potrebné menej úložného priestoru. Toto je užitočné pre používateľov, ktorí chcú rýchlo spustiť uzol a priamo interagovať so sieťou.

3.2 Solidity

Solidity je objektovo orientovaný, vysokoúrovňový jazyk pre implementáciu smart kontraktov. Smart kontrakty sú programy, ktoré riadia správanie účtov v rámci Ethereum stavu. Solidity je jazyk s kučeravými zátvorkami, cielený na Ethereum Virtual Machine (EVM). Je ovplyvnený jazykmi C++, Python a JavaScript. Je staticky typovaný, podporuje dedičnosť, knižnice a komplexné používateľom definované typy a mnoho ďalších vlastností. [12]

3.2.1 Remix

Remix je webový nástroj a integrované vývojové prostredie (IDE) určené pre vývoj smart kontraktov pre Ethereum blockchain. Umožňuje programátorom písať, testovať a implementovať smart kontrakty pomocou jazyka Solidity priamo v prehliadači. Remix poskytuje rôzne nástroje a funkcie na vývoj a ladenie smart kontraktov, vrátane editora kódu, kompilátora Solidity, simulátora a možnosti nasadenia na blockchain. Je to užitočný nástroj pre vývojárov, ktorí pracujú s Ethereum blockchainom a chcú rýchlo vytvárať a testovať smart kontrakty bez potreby zložitejších nástrojov alebo prostredia.

3.3 IPFS

IPFS je modulárny súbor protokolov na organizovanie a prenos dát, navrhnutý od základov s princípmi adresovania obsahu a peer-to-peer sietí. Pretože IPFS je open-source, existujú viaceré implementácie. Aj keď má viac ako jedno použitie, jeho hlavným použitím je publikovanie dát (súborov, adresárov, webových stránok atď.) v decentralizovanom štýle [13]. Termín IPFS môže referovať viacero konceptov:

- Konkrétnu implementáciu IPFS protokolu
- Decentralizovanú sieť zloženú z uzlov IPFS, ktorá je otvorená a participatívna
- .Modulárnu sadu protokolov a štandardov na organizovanie a prenos dát s obsahovou adresáciou. [13]

Aj keď má podobnosti s uvedenými systémami a často sa používa architektúrach s týmito systémami, IPFS nie je poskytovateľ úložiska. Napriek tomu že existujú poskytovatelia úložiska s podporou IPFS, IPFS je protokol a nie poskytovateľ. Títo poskytovatelia sa označujú ako takzvané "pinning services".

Výhody

- **Verifikovateľnosť** – IPFS používa kryptografické hashovanie na overenie integrity súborov. [14]
- **Odolnosť** - IPFS nemá jediný bod zlyhania a používatelia si nemusia navzájom dôverovať. Sieť bude funkčná, pokiaľ bude dostupný aspoň jeden bod. [14]
- **Výkon** - IPFS poskytuje rýchlejší prístup k údajom tým, že ich umožňuje replikovať a získavať z viacerých miest. Taktiež umožňuje používateľom pristupovať k údajom z najbližšieho miesta pomocou adresovania obsahu namiesto adresovania založeného na polohe. [14]
- **Link rot** - IPFS odstraňuje problém zneprístupnenia odkazov tým, že umožňuje adresovanie údajov podľa ich obsahu, a nie podľa ich umiestnenia. Inými slovami, obsah v IPFS je stále dostupný bez ohľadu na jeho umiestnenie a nezávisí od dostupnosti konkrétnych serverov. [14]
- **Off chain úložisko** – IPFS umožňuje overiteľné off chain úložisko vytvorením prepojenia medzi stavom blockchainu a adresovaným obsahom zverejneným na IPFS. Funguje to tak, že sa v smart kontrakte uloží identifikátor obsahu (CID). [14]

3.3.1 Pinata

Pinata je cloudová služba určená na ukladanie a správu digitálnych obsahov pomocou protokolu IPFS. Jej primárnym zámerom je poskytnúť užívateľom jednoduchý a spoľahlivý spôsob ukladania dát do decentralizovaného a odolného úložného systému. Pinata umožňuje užívateľom nahrávať a ukladať súbory, ako sú obrázky, videá, dokumenty a webové stránky. Tieto súbory sú potom distribuované pomocou IPFS. [15]

Služba Pinata poskytuje rôzne funkcie a nástroje na správu uložených súborov, vrátane možnosti prispôsobiť konfiguráciu ukladania a zdieľania súborov, sledovanie histórie verzií a monitorovanie prístupu k súborom. Jej cieľom je poskytnúť užívateľom jednoduché a efektívne riešenie pre ukladanie a zdieľanie digitálnych obsahov v decentralizovanom úložisku. [15]

3.4 React

React je moderná open-source JavaScript knižnica, ktorá sa používa na vytváranie užívateľských rozhraní a jednostránkových aplikácií (SPA). Je vyvíjaná a udržiavaná spoločnosťou Meta a je populárna medzi vývojármi pre svoju efektívnosť a flexibilitu. Hlavným konceptom Reactu je komponentová architektúra, ktorá umožňuje vytvárať znovu použiteľné a izolované časti kódu nazývané komponenty. Každý komponent je malá, samostatná jednotka, ktorá obsahuje svoj vlastný stav a zobrazenie. Tieto komponenty sa potom môžu zlúčiť do zložitejších štruktúr a zabezpečiť ľahkú údržbu a rozšíriteľnosť aplikácií. [16]

Jedným z hlavných dôvodov popularity Reactu je jeho schopnosť efektívne aktualizovať užívateľské rozhrania bez obnovenia celej stránky. React používa virtuálny DOM (Document Object Model), ktorý umožňuje optimalizovať výkon a rýchlosť webových aplikácií tým, že minimalizuje počet potrebných úprav v reálnom DOM. Okrem toho React podporuje jednosmerný dátový tok, ktorý umožňuje jednoduché a jasné riadenie dát a stavu aplikácie. To zjednodušuje debugovanie a testovanie aplikácií a zabezpečuje konzistentný stav medzi rôznymi časťami aplikácie. [16]

Celkovo je React silným nástrojom pre vývoj moderných webových aplikácií a poskytuje vývojárom robustné a efektívne riešenie pre tvorbu dynamických a interaktívnych užívateľských rozhraní. [16]

3.4.1 Wagmi

Wagmi je React Hook knižnica, ktorá adresuje viaceré problémy pri vývoji aplikácií na platforme Ethereum. Tieto problémy zahŕňajú komplexné aspekty pripojenia k rôznym peňaženkám, podporu pre viaceré blockchainy, zabezpečenie podpisovania správ a dát, odosielanie transakcií, sledovanie udalostí a zmien stavu, obnovenie zastaranej blockchainovej databázy a mnoho ďalších. Wagmi rieši tieto problémy a viac, umožňujúc vývojárom zamerať sa na vytváranie vysoko kvalitných a výkonných zážitkov pre platformu Ethereum. Ponúka pre vývojárov modulárne a komponované API, automatickú typovú kontrolu a podrobnú dokumentáciu. Výkonnosť je kľúčová pre aplikácie všetkých veľkostí, a preto je Wagmi optimalizované pre minimalizáciu veľkosti balíka pre rýchle načítanie stránok. Navyše má širokú podporu pre najpopulárnejšie funkcie Etherea. Je neustále aktualizované a vylepšované tímom vývojárov. [17]

3.5 MS Azure

Microsoft Azure (predtým známy ako Windows Azure) je cloudová platforma poskytovaná spoločnosťou Microsoft. Je to jeden z najväčších poskytovateľov cloudových služieb na svete, ktorý umožňuje organizáciám vytvárať, nasadzovať a spravovať rôznorodé aplikácie a služby v cloude. Poskytuje širokú škálu cloudových služieb, vrátane výpočtu, úložiska, dátových služieb, umelé inteligencie, internetu vecí (IoT), analýz a mnoho ďalších. Medzi hlavné vlastnosti a funkcie patrí elastický výpočet, škálovateľné úložisko, vysoká dostupnosť, bezpečnosť a možnosť integrácie s existujúcimi aplikáciami a infraštruktúrou. [18]

Azure umožňuje organizáciám efektívne využívať cloudové zdroje a prispôsobiť svoje riešenia podľa potrieb a požiadaviek. Je to ideálna platforma pre vývoj, testovanie, nasadenie a správu aplikácií a služieb bez nutnosti investícií do vlastnej hardvérovej infraštruktúry. Organizácie ho môžu využívať na rôzne účely, vrátane hostingu webových aplikácií, ukladania a spracovania dát, vývoja a nasadenia aplikácií na mobilných zariadeniach, implementácie umelých inteligentných a strojového učenia, a mnoho ďalších. [18]

3.5.1 DevOps

Azure DevOps je cloudová služba poskytovaná spoločnosťou Microsoft, ktorá zjednocuje ľudí, procesy a nástroje pre vývoj softvéru. Táto platforma poskytuje riešenia pre správu vývoja softvéru a jeho dodanie v cloude. Zahŕňa širokú škálu nástrojov a služieb, ktoré umožňujú tímom zlepšiť spoluprácu, automatizovať procesy a dodávať softvér rýchlejšie a spoľahlivejšie [19]. Medzi hlavné komponenty Azure DevOps patria:

- **Azure Boards**: Služba pre sledovanie práce a riadenie projektov, ktorá poskytuje agilný, zásadami riadený prístup k plánovaniu a sledovaniu práce.
- **Azure Repos**: Repozitár kódu s podporou Git, ktorý umožňuje tímom spravovať zdrojový kód, vykonávať verziovanie a spolupracovať na zmenách kódu.
- **Azure Pipelines**: Nástroj pre automatizované zostavovanie, testovanie a nasadenie softvéru na rôzne ciele, vrátane cloudových a lokálnych prostredí.
- **Azure Test Plans**: Služba pre plánovanie, sledovanie a správu testov, ktorá umožňuje tímom efektívne testovať softvér a identifikovať problémy.
- **Azure Artifacts**: Nástroj na správu a distribúciu balíčkov, ktorý poskytuje centralizované úložisko pre rôzne typy artefaktov. [19]

Azure Pipelines

Jedná sa o nástroj na automatizáciu procesov zostavenia, testovania a nasadzovania softvéru. Azure Pipelines umožňujú vytvárať a spravovať kontinuálny proces integrácie a dodania (CI/CD) pre aplikácie bežiace na rôznych platformách a v rôznych prostrediach [20]. Hlavné funkcie a možnosti Azure Pipelines zahŕňajú:

- **Flexibilita a konfigurovateľnosť**: Azure Pipelines podporujú rôzne programovacie jazyky, frameworky a nástroje, čo umožňuje tímom prispôsobiť si ich CI/CD procesy podľa ich potrieb.
- **Vizuálny dizajnér**: Tímom umožňuje jednoducho vytvárať a upravovať pipelines pomocou vizuálneho grafického rozhrania, čo zjednodušuje konfiguráciu a správu procesov.
- **Kód**: Konfigurácia sa uchováva ako kód a môže byť spravovaná a verziovaná pomocou systému Git. To umožňuje tímom efektívne sledovať zmeny a spolupracovať na procesoch CI/CD.
- **Integrácia s rôznymi nástrojmi a službami**: Azure Pipelines sú plne integrované s rôznymi ďalšími nástrojmi a službami, vrátane repozitárov kódu (napr. GitHub, Azure Repos), testovacích rámcov, cloudových platforiem a ďalších.
- **Škálovateľnosť a spoľahlivosť**: Sú navrhnuté tak, aby boli schopné zvládať aj veľké projekty a zabezpečovali spoľahlivé a rýchle vykonávanie.

Continuous Integration (CI) je praktika používaná vývojárskymi tímami na automatizáciu, zlúčenie a testovanie kódu. CI pomáha odhaľovať chyby v čo najskoršej fáze vývoja, čo ich robí menej nákladnými na opravu. Automatizované testy sa vykonávajú ako súčasť procesu na zabezpečenie kvality. CI systémy produkujú artefakty a poskytujú ich do procesov nasadzovania na podporu častých nasadení. [20]

Continuous Delivery (CD) je proces, pomocou ktorého sa kód zostavuje, testuje a nasadzuje do jedného alebo viacerých testovacích a produkčných prostredí. Nasadzovanie a testovanie v rôznych prostrediach zvyšuje kvalitu. Systémy CD produkujú nasaditeľné artefakty, vrátane infraštruktúry a aplikácií. Automatizované procesy používajú tieto artefakty na dodanie nových verzií a opráv existujúcich systémov. Systémy, ktoré monitorujú a zasielajú upozornenia, bežia neustále, aby zabezpečili prehľad nad celým procesom CD. [20]

4 Implementácia

4.1 Návrh systému

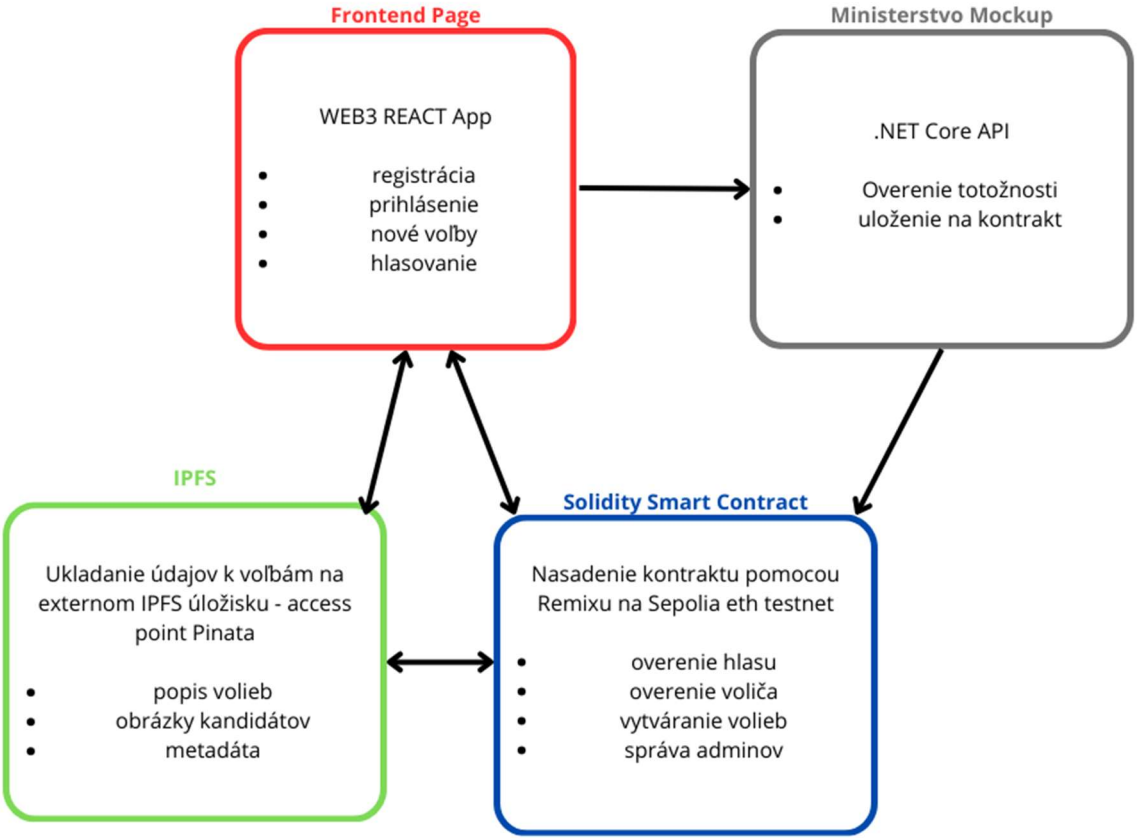
Ako prvé sme museli vykonať analýzu a navrhnúť systém. Aplikácia sa skladá z 4 hlavných častí, ktoré spolupracujú medzi sebou. Najdôležitejšou časťou je samotný smart kontrakt. Všetky nasledujúce časti sa odvíjajú od jeho implementácie. Nachádza sa tu zabezpečenie hlasovania a kontrola voliča. V kontrakte je možne taktiež aj vytvárať nové voľby. Táto funkcionalita by mala byť dostupná iba administrátorským používateľom. Smart kontrakt nebude nasadení na mainnet z dôvodu potreby reálneho Etheru. Na nasadenie zatiaľ postačí testovacia sieť Sepolia. Táto sieť používa Ether ktorý nemá reálnu hodnotu a dá sa bezplatne získať. Na celý proces vývoja a nasadenia je použitý webový nástroj Remix. Tento nástroj umožňuje rýchly výber Ethereum siete a následné nasadenie kontraktu na vybranú sieť.

Pre zníženie množstva dát uložených v smart kontrakte je použitá IPFS služba. Vďaka tomu bude ušetrené veľké množstvo poplatkov za gas. Budú sa tu nachádzať všetky metadáta o voľbách a kandidátoch, ktoré nie sú potrebné pre fungovanie hlasovania. Sú to iba doplnkové informácie ako meno, popis a obrázky. Pre túto funkcionalitu je použitá cloudová služba Pinata. Základné používanie tejto aplikácie je bezplatné a úplne postačí na ukážku funkčnosti hlasovania.

Ďalšou dôležitou časťou je zobrazenie všetkých dát, či už ide o proces hlasovania alebo dáta uložené na cloude pomocou IPFS. O túto funkcionalitu sa stará Front end webová stránka. Je napísaná v javascriptovom frameworku React. Táto knižnica obsahuje rozšírenie Wagmi ktoré zjednoduší prihlasovanie a celé fungovanie funkcií zo Smart kontraktu. Celý front end by mal slúžiť iba ako zobrazovač dát uložených na blockchaine. Celá logika hlasovania ako aj zabezpečenie sa nachádzajú na kontrakte. Zároveň by malo byť ľahké a prístupné nasadiť vlastný kontrakt podľa predlohy a používať ho na vlastné účely ako napríklad hlasovanie vo firme alebo škole.

Záverečnou časťou práce je vytvoriť náhradu overenia totožnosti užívateľa. Keďže nemáme prístup k reálnym dátam o občanoch a ich možnosti hlasovania, bude vytvorený falošný systém ktorý bude iba potvrdzovať totožnosť užívateľov. Tieto dáta nebudú nikde ukladané. Na webovej stránke sa bude nachádzať jednoduchý formulár, ktorý pošle

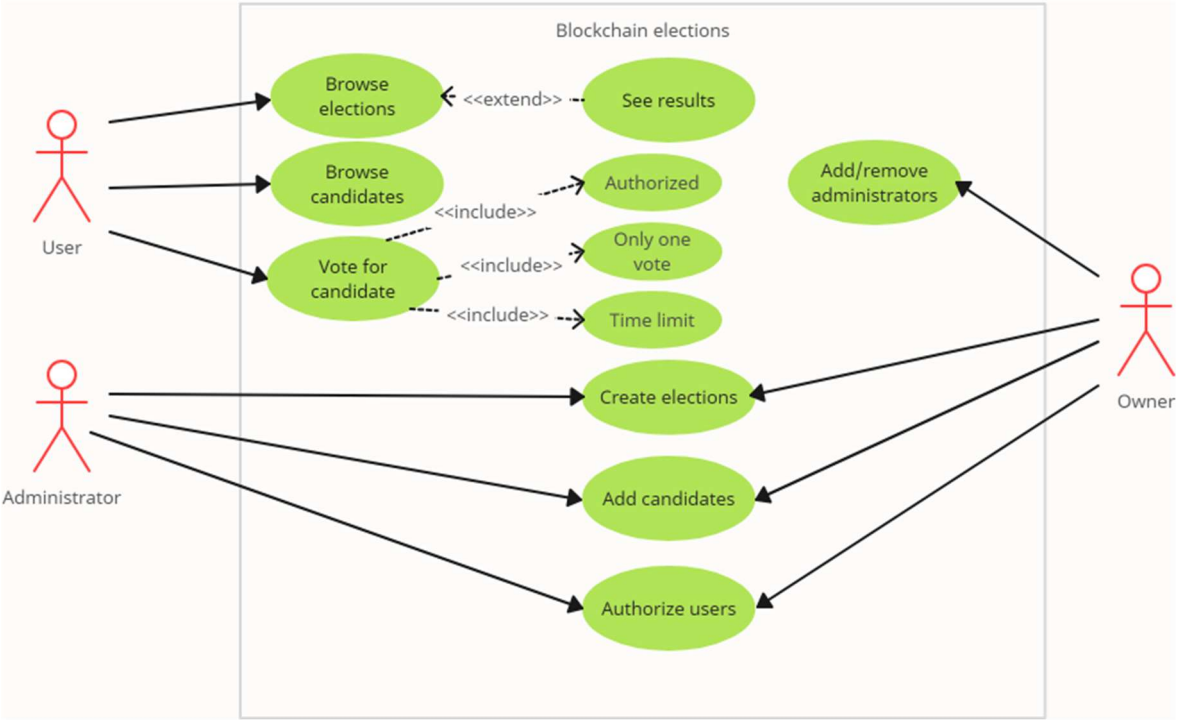
požiadavku na Backend server. Ten túto požiadavku spracuje a zmení stav daného účtu v smart kontrakte na overený.



Obr. 5 Návrh systému

Na obrázku je možné vidieť grafickú reprezentáciu týchto 4 samostatných systémov. Každá časť spolupracuje iba s potrebnou časťou druhého systému. Napríklad Backend systém na overenie používateľov získava dáta z webovej stránky a následne zapisuje stav na smart kontrakt. Webová stránka ako získava, tak aj zapisuje dáta na smart kontrakt. To isté vykonáva aj pre cloud IPFS systém. Administrátorské rozhranie totižto umožňuje vytváranie nových volieb a kandidátov ktoré musia byť uložené na blockchain a metadáta k ním na túto IPFS službu.

4.1.1 Use Case Diagram



Obr. 6 Use Case Diagram

Na obrázku vyššie sa nachádza diagram prípadov použitia pre online hlasovanie. Aplikácia obsahuje tri typy používateľov. Prvý z nich je klasický "user", ktorý môže čítať všetky dáta uložené na blockchaine a taktiež vidí históriu celej blockchain siete. Vďaka tomu pozná výsledky hlasovania. Jeho jedinou vykonateľnou akciou, ktorá mení stav kontraktu je hlasovanie za kandidáta. V niektorých prípadoch musí byť užívateľ overený. Taktiež sa overuje aby v jednom hlasovaní mohol hlasovať maximálne jeden krát a to iba v správnom časovom intervale.

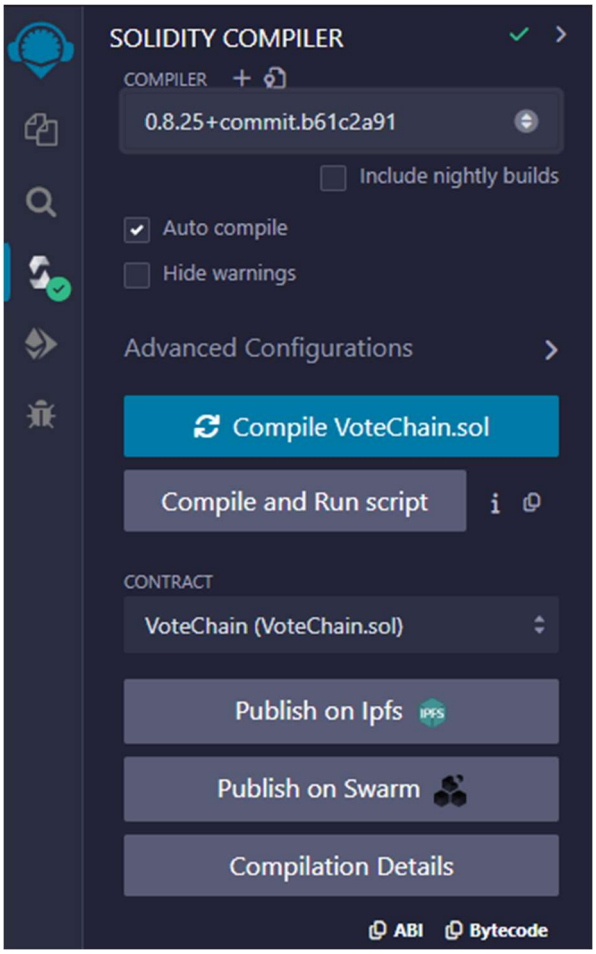
Vlastník a administrátor zdieľajú viaceré funkcionality na správu hlasovania. Dokážu vytvárať nové voľby. V nich môžu pridávať kandidátov. Taktiež dokážu autorizovať používateľov. Vlastník ešte navyše k tomu môže pridať alebo odstrániť administrátora.

4.2 Smart kontrakty

Vytváranie kontraktov prebiehalo postupne. Začalo sa jednoduchým kontraktom, ktorý dokázal iba ukladať jednoduché premenné a meniť ich stav. V každej iterácii boli zmenené alebo pridané nové funkcionality. Celý tento proces bol možný kvôli použitiu webovej aplikácie Remix.

4.2.1 Remix

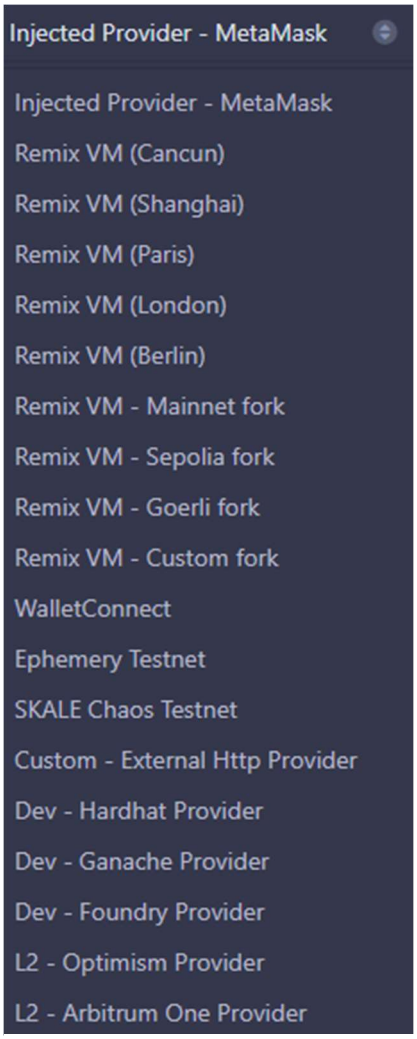
Táto aplikácia obsahuje všetky potrebné funkcie na rýchly vývoj smart kontraktov. Napísaný kontrakt skompiluje a overí, či sa v ňom nenachádzajú chyby. Pred týmto procesom je ale najskôr dôležité vybrať verziu kompilátora. Pre túto prácu bola použitá v tej dobe najnovšia verzia "0.8.25+commit.b61c2a91". Každá verzia kompilátora pridáva nové zmeny, čiže je dôležité používať stále tú istú verziu pre ktorú bol kontrakt napísaný. Remix taktiež dokáže pretvoriť skompilovaný kód na ABI. Toto ABI bude dôležité v časti ktorá sa venuje Front endu pri volaní funkcií pomocou Wagmi.



Obr. 7 Remix možnosti kompilátora

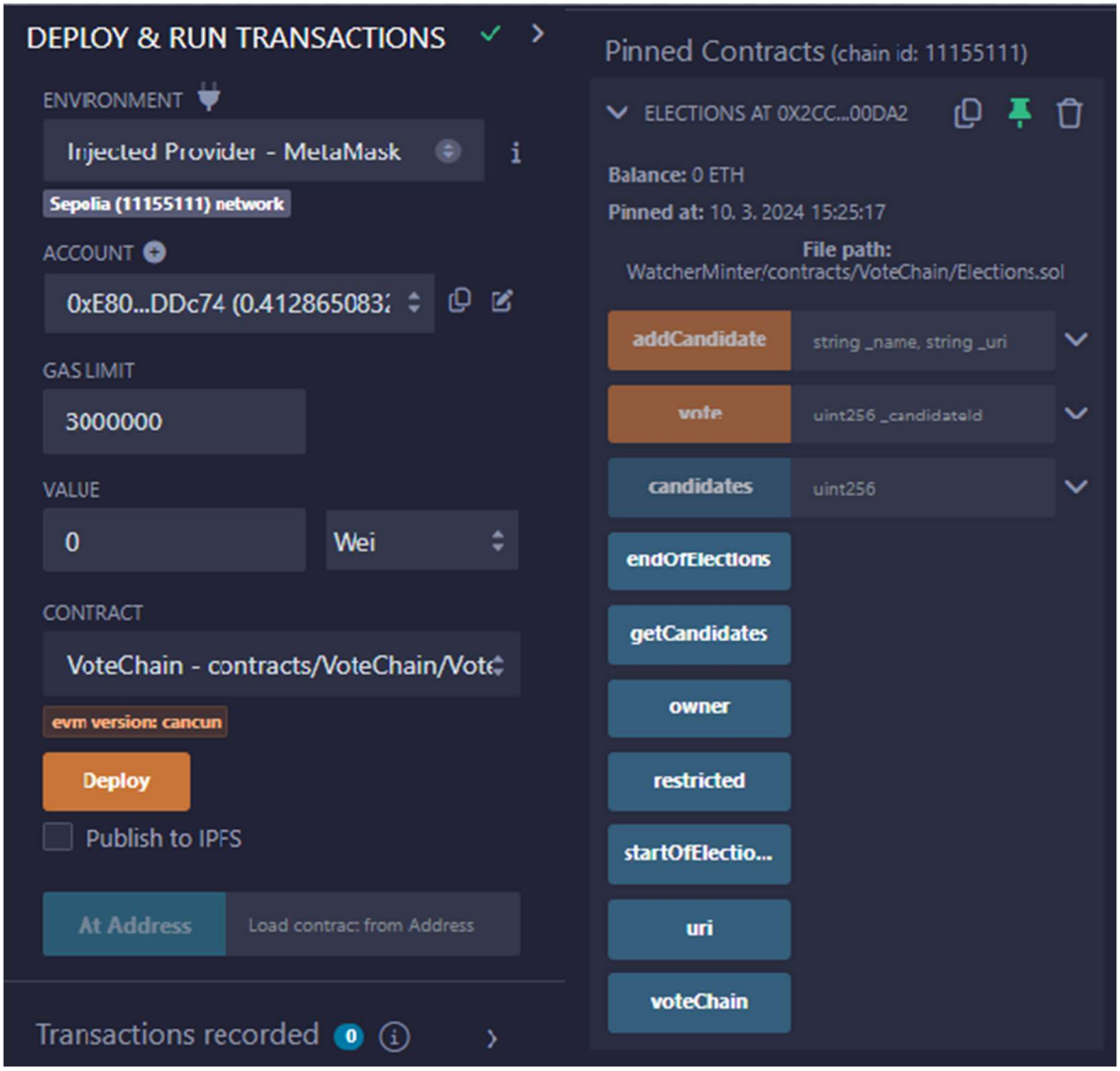
Skompilovaný kontrakt je následne možné nasadiť. Na výber je hneď niekoľko možností. Na prvotný proces vývoja kontraktu je najlepšie použiť Remix VM. V ňom sa nachádzajú rôzne oblasti, ako napríklad Berlin, London, Mainnet fork alebo Sepolia fork. Po výbere niektorej Remix VM siete máme dostupných 15 testovacích účtov. Na každom z nich sa nachádza 100 etheru. Po dokončení fázy vývoja môžeme vybrať iný typ siete, ktorý umožní nasadenie kontraktu na oficiálnu sieť Sepolia alebo Mainnet. Žiadna z týchto sieti

ale nie je vo výbere dostupná, pretože sa musíme prihlásiť pomocou peňaženky. Vyberieme si Injected provider alebo Wallet Connect. Pred tým ale musíme mať už vytvorený účet na niektorej z týchto možností. Injected provider umožňuje prihlásenie pomocou prehliadačového rozšírenia MetaMask. Momentálne je to jeden z najrozšírenejších spôsobov prihlasovania sa a správy Ethereum účtov. Remix automaticky zistí, na ktorej sieti je peňaženka prihlásená a implementuje ju do rozhrania. To znamená, že Sepoliu sieť si najskôr musíme zvoliť v MetaMask aplikácii. Na nasadenie kontraktu pomocou tejto možnosti je nutné vlastniť daný typ Etheru. Ako bolo spomenuté v predchádzajúcej časti, Sepolia ETH môžeme získať pomocou takzvaných faucetov.



Obr. 8 Výber siete v Remixe

Nasadenie kontraktu už prebieha veľmi jednoducho. Ak kontrakt obsahuje nejaký konštruktor s parametrami, tie stačí už iba vyplniť. Danú transakciu je potrebné potvrdiť a po pár sekundách sa kontrakt nasadí na konkrétnu adresu. Taktiež je možné sa pripojiť na už nasadení kontrakt. Stačí poznať jeho adresu. Po nasadení alebo pripojení sa zobrazí výber všetkých dostupných metód kontraktu. Toto rozhranie umožňuje veľmi efektívne testovanie. Môžeme volať akúkoľvek metódu, nastaviť limit na poplatok za gas alebo debugovať. Oranžové metódy zapisujú alebo inak upravujú kontrakt, ich vykonávanie trvá dlhšiu dobu kvôli čakaniu na blockchain. Modré iba čítajú zapísané dáta a nepotrebujú potvrdenie transakcie. V konzole sa zobrazujú všetky dôležité informácie o transakciách na blockchaine.



Obr. 9 Nasadenie kontraktu

4.2.2 VoteChain a Elections smart kontrakty

Finálna verzia pracuje s dvoma kontraktmi. Prvý kontrakt spravuje autorizáciu voličov a ich administráciu. Jeho hlavnou úlohou je ale zoskupovať voľby. Tento kontrakt budeme volať VoteChain. Druhý kontrakt je iba predlohou na voľby. Nazýva sa Elections. Pre každé voľby je potrebné nasadiť nový Elections kontrakt a priradiť jeho správu hlavnému VoteChain kontraktu.

VoteChain kontrakt

Ako už bolo spomenutú, VoteChain kontrakt spravuje autorizáciu voličov. Obsahuje mapping(address => bool) isAuthorizedVoter. Mapovanie v Solidity funguje podobne ako dátová štruktúra hashmap v iných programovacích jazykoch. Mapuje unikátny kľúč na určitú hodnotu. V solidity sa nachádza špeciálna dátová štruktúra address. Je to 20 bitová hodnota, ktorá reprezentuje Ethereum adresu. Konkrétne toto mapovanie priraďuje účtu na Ethereu hodnotu true alebo false na základe toho, či bol užívateľ overený. Pre túto hodnotu bola vytvorená get metóda ktorá je použitá v Elections kontrakte.

```
1  // SPDX-License-Identifier: GPL-3.0
2
3  pragma solidity >=0.7.0 <0.9.0;
4
5  contract VoteChain {
6
7      mapping(address => bool) isAuthorizedVoter;
8
9      address public owner;
10
11     mapping(address => bool) public isAdmin;
12
13     address[] elections;
14
15
16     constructor() {    514790 gas 490000 gas
17         owner = msg.sender;
18     }
```

Obr. 10 Atribúty VoteChain kontraktu

Ďalej sa tu nachádza adresa vlastníka kontraktu ktorá sa automaticky nastaví v konštruktore. Iba vlastník kontraktu môže pridávať a odstraňovať administrátorov. Ich zoznam je verejný a nachádza sa v rovnakej štruktúre ako autorizovaný voliči. Administrátorom sa môže stať aj iný kontrakt.

Posledným atribútom je pole adries Elections kontraktov. Toto nám umožní rýchle prehľadávanie všetkých volieb priradených správcovskému kontraktu.

```
modifier adminOnly() {
    require(msg.sender == owner || isAdmin[msg.sender] == true);
    _;
}

modifier ownerOnly() {
    require(msg.sender == owner);
    _;
}
```

Obr. 11 VoteChain modifikátory

Solidity obsahuje modifikátory. Sú to funkcie, ktoré môžu zmeniť správanie alebo stav kontraktu. Používajú sa na zjednodušenie a zjednotenie opakujúcich sa vzorov v kóde. Modifikátory môžu byť priradené k funkciám a umožňujú vykonať určité úkony pred alebo po spustení danej funkcie. VoteChain kontrakt obsahuje takzvané modifikátori prístupu. Zabezpečujú, že funkcie môžu byť volané iba určitými adresami ako napríklad administrátori alebo vlastník.

```
46   function unauthorizeVoter(address _address) external adminOnly {   29292 gas
47       isAuthorizedVoter[_address] = false;
48   }
49
50   function isAuthorized(address _address) public view returns (bool){   2971 gas
51       return isAuthorizedVoter[_address];
52   }
```

Obr. 12 Ukážka funkcií v solidity

Na obrázku môžeme vidieť dva príklady deklarácie funkcií v Solidity. Prvá funkcia obsahuje modifikátor adminOnly. Ako už bolo spomenuté, kvôli nemu túto funkciu môže volať iba administrátori definovaný v mapovaní isAdmin. Funkcia je označená ako external. To znamená, že nemôže byť volaná vo vnútri tohto kontraktu a môže ju volať iba externý kontrakt alebo účet. Taktiež táto funkcia neobsahuje žiadnu návratovú hodnotu. Pri jednoduchých funkciách, Remix dokáže odhadnúť cenu danej funkcie. Pri písaní kontraktov je veľmi dôležité optimalizovať túto hodnotu.

Druhá funkcia je označená ako public, to znamená že môže byť volaná odkiaľkoľvek. Ďalšia hodnota view je veľmi dôležitá. Toto kľúčové slovo sa používa na označenie funkcií, ktoré čítajú údaje zo stavu kontraktu alebo z blockchainu, ale neprodukujú žiadne zmeny

stavu. View funkcie nevyžadujú žiadne transakčné poplatky na ich volanie, pretože nezmenia stav kontraktu alebo siete. Remix ale aj tak odhadol hodnotu volania tejto funkcie. Treba si dať pozor na to, že poplatky za gas sa platia aj pri view funkciách, ak sú volané inými funkciami. Solidity obsahuje ešte dva podobné modifikátori payable a pure. Payable funkcie nás nebudú zaujímať, pretože označujú také funkcie, ktoré môžu prijímať Ether od používateľov v rámci transakcií. Pure funkcie nečítajú žiadne údaje zo stavu kontraktu ani z blockchainu a nevyžadujú žiadne vstupy, ale iba vykonávajú výpočty na základe svojich parametrov. Sú úplne deterministické a ich výstup závisí iba od vstupných parametrov. Tieto funkcie sa používajú na vykonávanie matematických alebo iných výpočtov. Taktiež nevyžadujú žiadne transakčné poplatky ako view funkcie.

Elections kontrakt

Elections kontrakt je omnoho zložitejší. Obsahuje celú logiku hlasovania a overovania.

```
5   contract Elections {
6       struct Candidate {
7           uint id;
8           string name;
9           string uri;
10          uint256 voteCount;
11      }
12
13      address public owner;
14
15      VoteChain public voteChain;
16
17      string public uri;
18
19      bool public restricted;
20
21      //Date of ending
22      uint256 public endOfElections;
23      //Date of start
24      uint256 public startOfElections;
25
26      Candidate[] public candidates;
27
28      mapping(address => bool) voted;
29
```

Obr. 13 Atribúty Elections kontraktu

V jazyku Solidity, struct je šablóna alebo štruktúra, ktorá umožňuje programátorom definovať vlastné dátové typy, ktoré môžu obsahovať rôzne typy údajov zhlukovaných do jedného celku. Kontrakt Elections definuje štruktúru Candidate ktorá obsahuje jeho id, meno, uri a počet hlasov. V kontrakte sa nachádza atribút voteChain typu VoteChain. Táto deklarácia umožňuje volať metódy tohto správcovského kontraktu. Ďalší atribút uri je podobný ako sa nachádza v štruktúre kandidáta. Do tejto hodnoty sa vkladá CID z cloudového úložiska IPFS služby. Pomocou takejto služby uložíme JSON súbor na cloud, ktorý bude obsahovať dodatočné parametre ako napríklad názov, popis a fotku volieb alebo kandidáta. Týmto spôsobom ušetríme obrovské množstvo poplatkov potrebných na ukladanie dát na blockchain a zároveň nestratíme decentralizovanú vlastnosť aplikácie.

Parameter restricted označuje, či sa volieb môžu zúčastniť iba overený voliči, ktorý sa nachádzajú v správcovskom kontrakte VoteChain. Kontrakt taktiež určuje časové okno. Iba v ňom je možné hlasovať. Solidity nemá žiadnu špecifickú dátovú štruktúru na ukladanie dátumu a času. Preto sa používa unixový čas. Je to bežný spôsob reprezentácie času v mnohých systémoch. Je to počet sekúnd, ktoré uplynuli od 1. januára 1970. Pre zistenie aktuálneho času sa používa metóda block.timestamp. Ďalej sa tu nachádza už iba pole kandidátov a mapovanie voličov, či už hlasovali. Takto zamedzíme viacnásobným pokusom hlasovať tým istým používateľom.

```
29
30      constructor(    infinite gas 1113600 gas
31          address _voteChainAddress,
32          string memory _uri,
33          uint256 _startOfElections,
34          uint256 _endOfElections,
35          bool _restricted
36      ) {
37          voteChain = VoteChain(_voteChainAddress);
38          uri = _uri;
39          owner = msg.sender;
40          startOfElections = _startOfElections;
41          endOfElections = _endOfElections;
42          restricted = _restricted;
43      }
```

Obr. 14 Konštruktor Elections kontraktu

Pomocou konštruktora sú nastavené všetky parametre. Je tu potrebné definovať aj správnu adresu VoteChain kontraktu aby bolo možné volať jeho metódy. Správne poradenie nasadenia kontraktov je nasledovné: Ako prvý vytvoríme VoteChain kontrakt a uložíme si jeho adresu. Vytvoríme JSON súbor s metadátami a uložíme ho pomocou IPFS cloudovej služby. CID tohto JSON súboru a adresu VoteChain kontraktu následne použijeme v konštruktore Elections smart kontraktu. Adresu tohto nového kontraktu použijeme vo funkcii AddElections vo VoteChain. Takto sa oba kontrakty navzájom previažu. Týmto spôsobom môžeme pridať ľubovoľné množstvo hlasovaní. Front end rozhranie bude obsahovať grafické rozhranie na automatické vykonanie tohto procesu.

```
71
72  abstract contract VoteChain {
73      function isAuthorized(address _address) public view virtual returns (bool);    - gas
74  }
```

Obr. 16 Definovanie abstraktného rozhrania kontraktu

Aby sme mohli volať metódy iného kontraktu, nestačí nám ho iba pridať ako atribút. Pre túto funkcionalitu je potrebné definovať rozhranie s danými funkciami, ktoré máme záujem používať. V našom prípade postačí jediná metóda VoteChain kontraktu. IsAuthorized vráti booleovskú hodnotu, či je konkrétny užívateľ autorizovaný na hlasovanie. Samozrejme sa to týka iba takých volieb, ktoré sú označene ako "restricted". Túto hodnotu vracia z mapovania. Pre bežné účely by postačilo označiť mapovanie ako public a to by sprístupnilo rovnakú hodnotu. Tento prípad ale nie je možné použiť v definovaní rozhrania kontraktu a preto bolo nutné vytvoriť "get" metódu.

```
function vote(uint256 _candidateId) external {    infinite gas
    require(_candidateId < candidates.length, "Candidate does not exist");
    if (restricted) {
        require(voteChain.isAuthorized(msg.sender), "You are not authorized");
    }
    require(block.timestamp > startOfElections, "Elections not starded yet");
    require(block.timestamp < endOfElections, "Elections already ended");
    require(!voted[msg.sender], "You already voted");
    voted[msg.sender] = true;
    ++(candidates[_candidateId].voteCount);
}
```

Obr. 15 Hlasovacia funkcia

Metóda na hlasovanie neobsahuje žiadny modifikátor. Volať ju môže každý. Až v tele funkcie sa vyhodnocuje, či je volič oprávnení voliť a či jeho hlas bude započítaný. Najskôr prebieha kontrola, či volený kandidát existuje. Každá kontrola obsahuje aj chybovú hlášku, ktorú je možné zobraziť na webovej stránke. Vďaka tomu stačí kontrolovať správnosť hlasu iba v smart kontrakte. Pred zapísaním transakcie a zaplatením poplatkov za gas sa ale najskôr skontroluje či je metódu možné vykonať. To zabraňuje zbytočnému plateniu poplatkov za neuskutočniteľné volania metód a šetrí čas čakania na výsledok. Ďalej sa tu nachádza kontrola na autorizáciu voliča, ktorá prebieha iba v prípade že je hlasovanie označené ako restricted. Pomocou získania aktuálneho unix času vieme skontrolovať, či bol hlas poslaný iba v časovom intervale určenom na hlasovanie. Ako posledné je iba ošetrené aby volič mohol hlasovať maximálne jeden krát pre konkrétne voľby.

Zaujímavosťou je riadok kde sa inkrementuje počet hlasov. Ak chceme iba pridať hodnotu 1 k premennej v bežnom programovacom jazyku, nezáleží či použijeme i++ alebo

++i, jediný rozdiel je vracaná hodnota. Zatiaľ čo i++ vráti hodnotu pred inkrementáciou, ++i vráti už zväčšenú hodnotu. V našom prípade keď sa snažíme znížiť náklady za gas a každá operácia nás niečo stojí, je veľmi dôležité vybrať správny typ. Príkaz i++ sa kompiluje do niečoho ako :

```
j = i;
i = i + 1;
return j
```

Príkaz ++i sa kompiluje na :

```
i = i + 1;
return i;
```

Použitím ++i ušetríme malé množstvo poplatku za gas, keďže sa vykonáva menšie množstvo príkazov.

4.3 Front end

Na zobrazenie všetkých informácií a prácu s kontraktmi bola použitá webová aplikácia. Vďaka tomu bude aplikácia ľahko prístupná pre všetkých bez potreby inštalácie dodatočného softvéru. Ako už bolo spomenuté niekoľko krát, celú bezpečnosť a integritu zabezpečuje blockchain. Každá transakcia musí byť podpísaná privátnym kľúčom, čo zaručí že môže hlasovať iba taký užívateľ, ktorý má prístup k tomuto kľúču.

4.3.1 Vytvorenie projektu

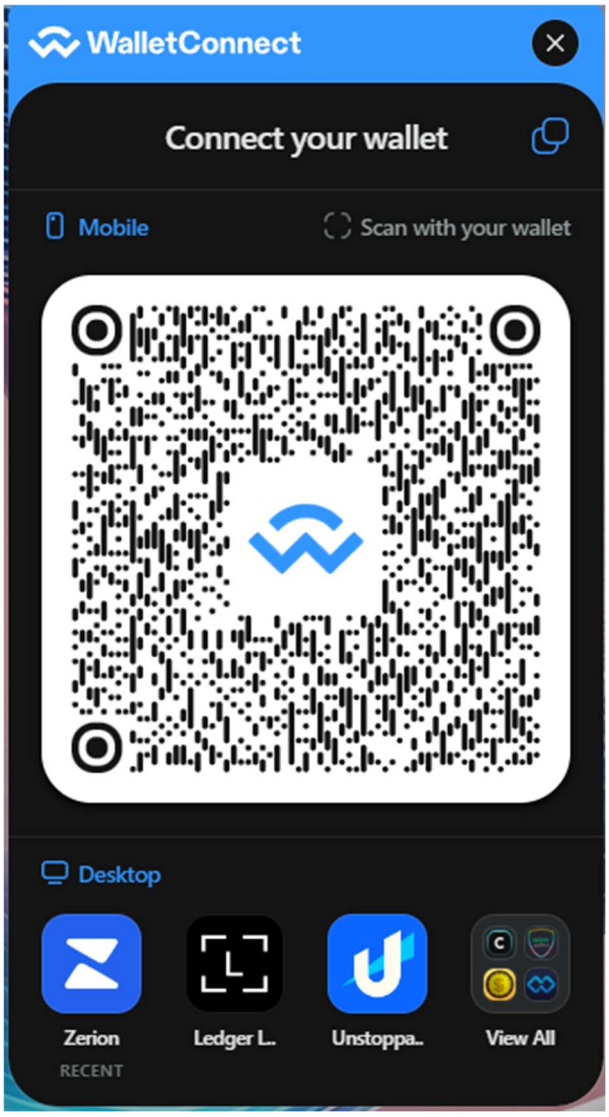
V tejto dobe sa na trhu nachádza veľké množstvo rôznych webových frameworkov. Jedným z najpoužívanejších je aj React. Rozhodli sme sa použiť práve túto knižnicu. Z hlavných dôvodov výberu bola jednoduchosť a rýchlosť vývoja softwaru. Taktiež je už vytvorených a otestovaných mnoho knižníc na prácu s blockchainom. Pravidelne sú tieto doplnky kontrolované a vychádzajú bezpečnostné opravy. React aplikácia bola vytvorená pomocou nástroja Vite. Je to moderný nástroj na vývoj webových aplikácií, ktorý je známy svojou vysokou rýchlosťou a jednoduchým používaním. Vite je o 10 až 100 krát rýchlejší ako ostatné JavaScriptové bundlery. To je preto, že je napísaný v jazyku Go a jeho závislosti sú pred zbierané s esbuild. Okrem toho Vite poskytuje zdrojový kód ako natívny ESM, čo umožňuje transformáciu a načítanie na požiadanie pre zlepšenú produktivitu vývojára. [21] Po vytvorení, aplikácia už obsahovala ukážkový príklad ktorý bolo potrebné vymazať.

4.3.2 Prihlasovanie

V bežných aplikáciách sa užívateľ prihlasuje pomocou názvu účtu a hesla. WEB3 aplikácie používajú úplne iný systém. Prihlasovanie prebieha pomocou Ethereum účtov. Tieto účty sú zvyčajne spravované nejakou krypto peňaženkou.

Obr. 17 Prihlasovanie

V našom prípade sú povolené dva najznámejšie spôsoby. Prvou z nich je MetaMask. Po zakliknutí tlačidla na prihlásenie MetaMaskou, aplikácia zistí či prehliadač obsahuje webový doplnok tejto peňaženky. Ak áno, vyskočí na požívateľa prihlasovacie okno z jeho MetaMask aplikácie. Tu musí vybrať ktorý účet chce použiť. Následne musí povoliť že aplikácia bude mať prístup k jeho Ethereum adrese, balancu, aktivite a bude môcť navrhovať transakcie na schválenie. Jedná sa o takzvané "injected" prihlásenie. Vďaka použitiu najnovších knižníc Wagmi je možné použiť aj ďalšie podobné typy prehliadačových peňaženiek pod touto "injected" kategóriou.



Obr. 18 Wallet Connect prihlásenie

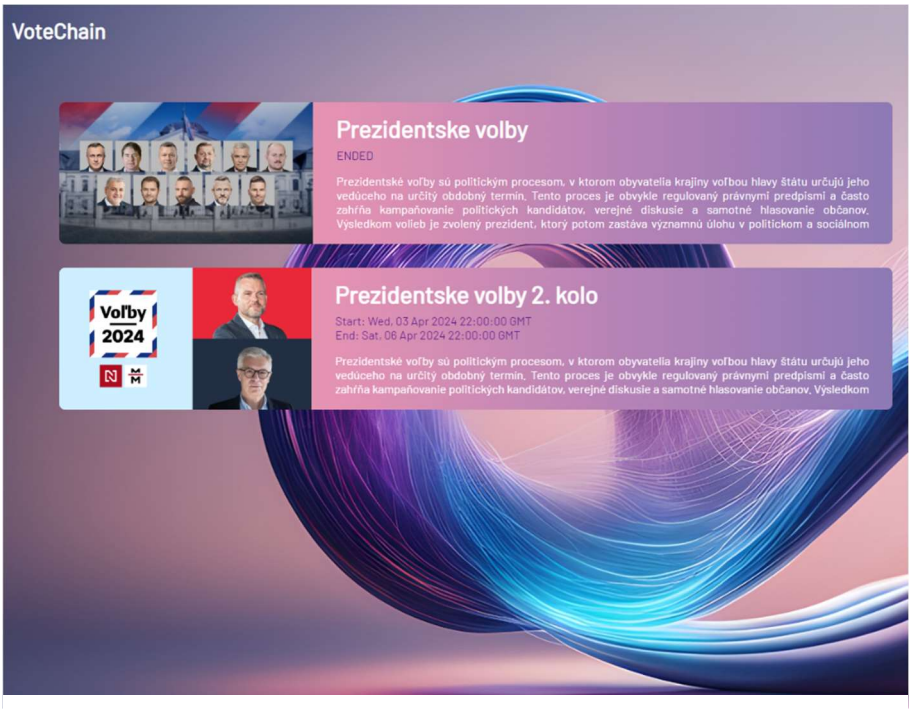
Druhým typom je Wallet Connect. Ak sa naša krypto peňaženka nachádza na smart mobilnom zariadení, potrebujeme použiť tento typ prihlásenia. Po kliknutí na tlačidlo sa zobrazí QR kód. Po naskenovaní a prihlásení sa na mobilnom zariadení nás aplikácia prihlási

aj v prehliadači. V najnovších verziách sa tu nachádzajú aj desktopové aplikácie pre peňaženky.

Na stránke sa nachádza aj jednoduchý návod na vytvorenie jednotlivých peňaženiek. Pri testovacej verzii aplikácie nasadenej na Sepolia Ethereum je potrebné pred hlasovaním doplniť aspoň malé množstvo SepoliaETH. Robí sa to pomocou faucetov. Na webe ich môžeme nájsť hneď niekoľko. Každý funguje trochu inak ale všetky potrebujú nejaký typ autentifikácie z dôvodu popularity testovacích sietí. Niektoré na doplnenie testovacieho Etheru vyžadujú aby účet vlastnil nejaké malé množstvo Mainnet Etheru. Iné faucety zase používajú ťaženie kryptomeny ako overovací faktor.

4.3.3 Zoznam dostupných hlasovaní

Po prihlásení webová stránka načíta všetky dostupné voľby v danom VoteChain kontrakte. V základe aplikácia obsahuje jeden preddefinovaný kontrakt ktorý bol použitý na testovacie účely. Používateľ má ale možnosť tento kontrakt zmeniť za svoj vlastný a pracovať s novými dátami. Vďaka tejto funkcionalite si každý užívateľ môže spravovať vlastné hlasovania napríklad v škole alebo vo firme. Po nasadení vlastného kontraktu sa automaticky stáva administrátorom a môže vytvárať hlasovania. Taktiež môže pridať nových administrátorov alebo overovať účty. Aplikácia sa snaží ponechať čo najväčšiu voľnosť pre užívateľa aby vyhovovala všetkým typom prostredia. Ďalej v každom hlasovaní dokáže pridávať kandidátov.

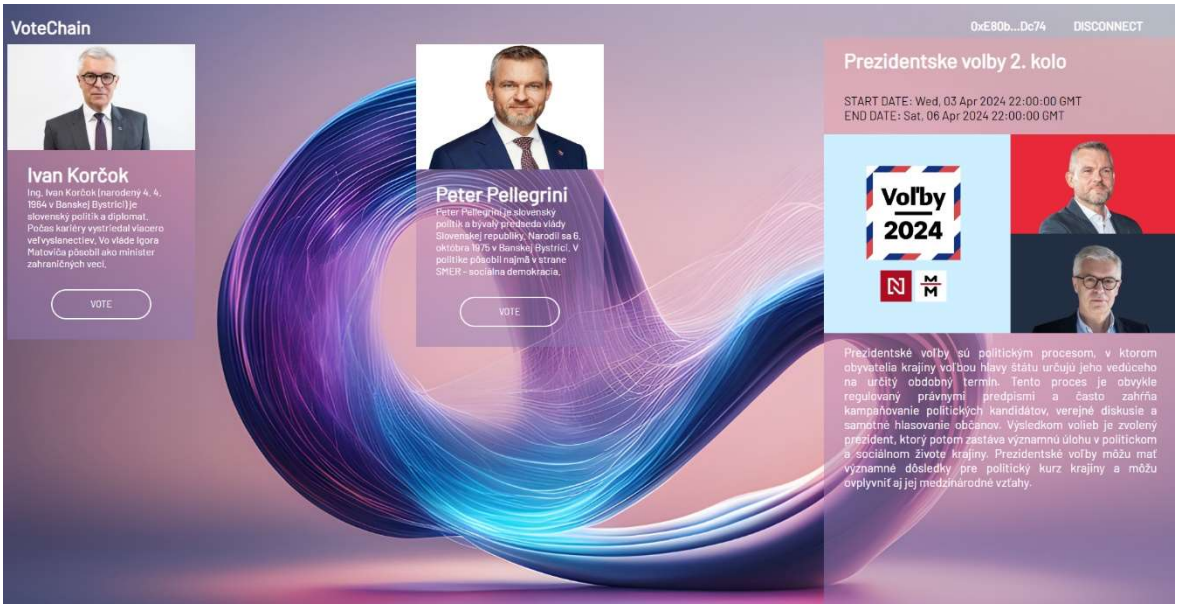


Obr. 19 Ukážka zoznamu dostupných hlasovaní

Na obrázku číslo 18. sa nachádza ukážka testovaného VoteChain kontraktu. Ku každým voľbám sa načítajú metadáta z cloudovej IPFS služby. Tieto metadáta sú uložené v JSON formáte. Nachádzajú sa tu tri položky: názov, popis a uri na obrázok. Podobne ako sa ukladá tento JSON do kontraktu pomocou CID, aj obrázok v ňom obsahuje iba CID odkaz na obrázok uložený na IPFS. Aplikácia taktiež prekladá unixový čas uložený na kontrakte do času prehliadača, ktorý používateľ používa. Ak čas priebehu hlasovania už skončil, označia sa iba ako skončené.

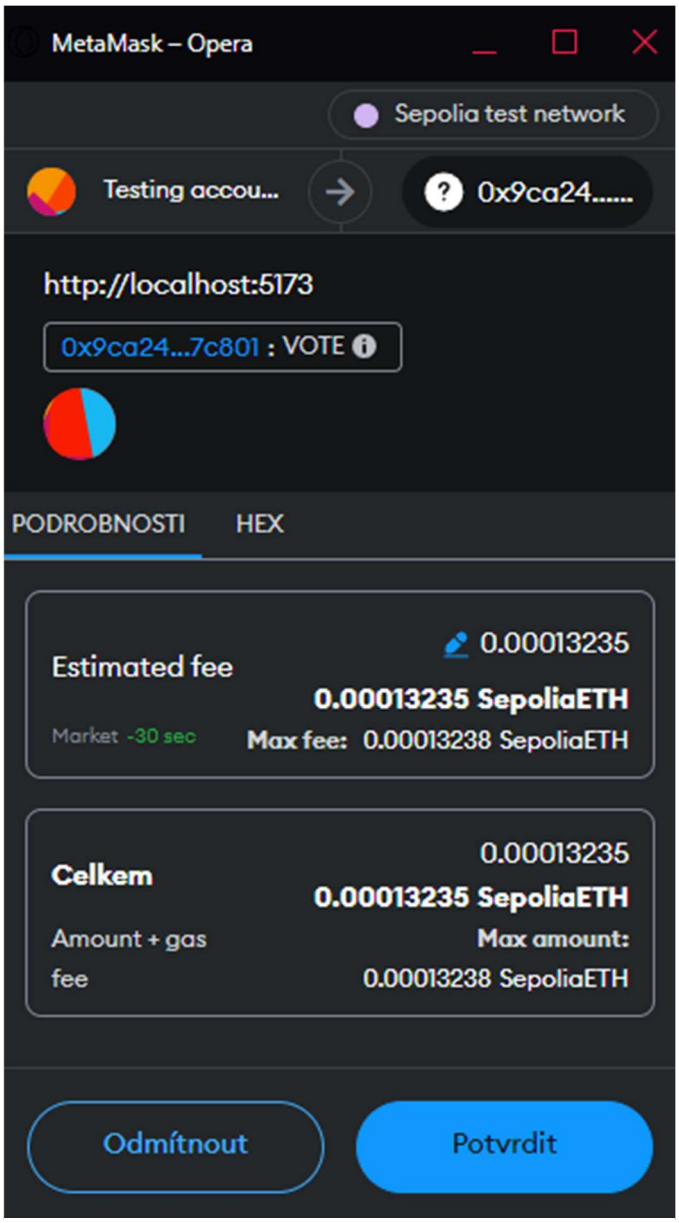
4.3.4 Voľby

Po kliknutí na prebiehajúce voľby sa zobrazí stránka so všetkými kandidátmi. Podobne ako pri zozname volieb aj tu je obrázok, meno a popis kandidátov uložený na cloudovej IPFS službe. Sú to statické dáta ktoré sa nemenia a ani neslúžia na žiadnu funkcionalitu v kontrakte. Na bočnom paneli sa nachádzajú všetky informácie o hlasovaní.



Obr. 20 Ukážka stránky hlasovania

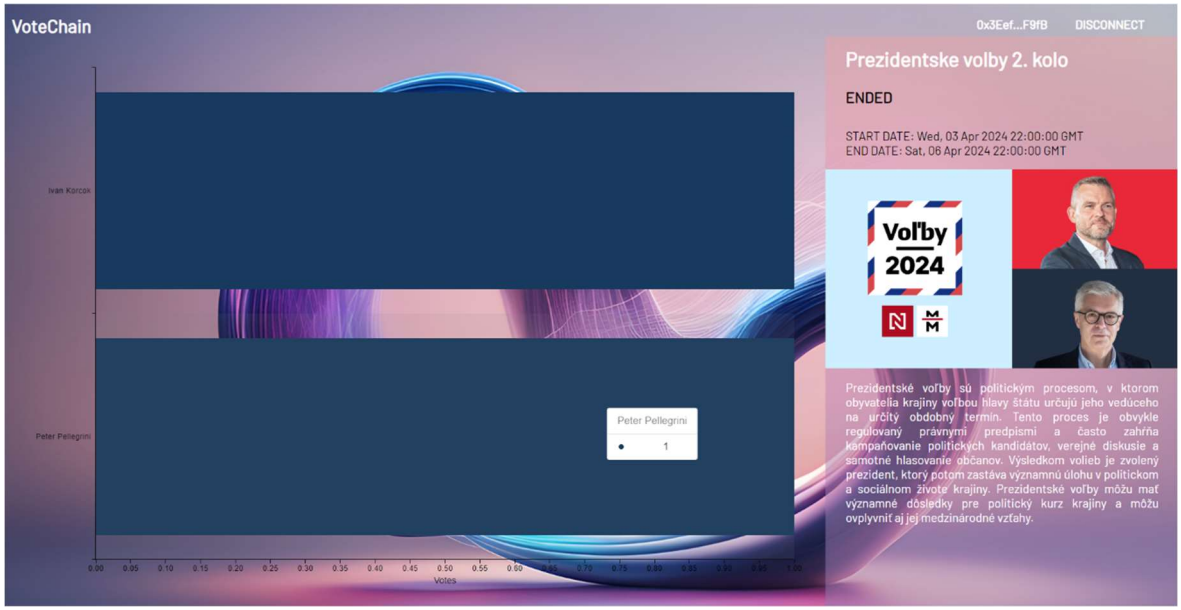
Po kliknutí na hlasovanie za kandidáta sa zobrazí ešte overenie, či si je užívateľ naozaj istý svojím rozhodnutím. Po tomto potvrdení je ešte potrebné schváliť transakciu. Tu záleží na type prihlásenia. Celé toto potvrdenie a výpočet poplatku za gas rieši aplikácia ktorou sme sa prihlásili. Napríklad v prípade MetaMasky sa zobrazí na kraji obrazovky kontextové okno. Tu sa nachádza aktuálny odhad ceny poplatkov. Taktiež je tu aj informácia o transakcii čakajúcej na potvrdenie, čiže užívateľ si môže skontrolovať či je to tá správna transakcia s dobrými parametrami. Po schválení aplikácia ešte informuje užívateľa, že bol hlas započítaný. V MetaMask aplikácii si ešte užívateľ môže skontrolovať všetky schválené transakcie a či už boli spracované a zapísané na blockchain.



Obr. 21 MetaMask potvrdenie transakcie

Výsledky

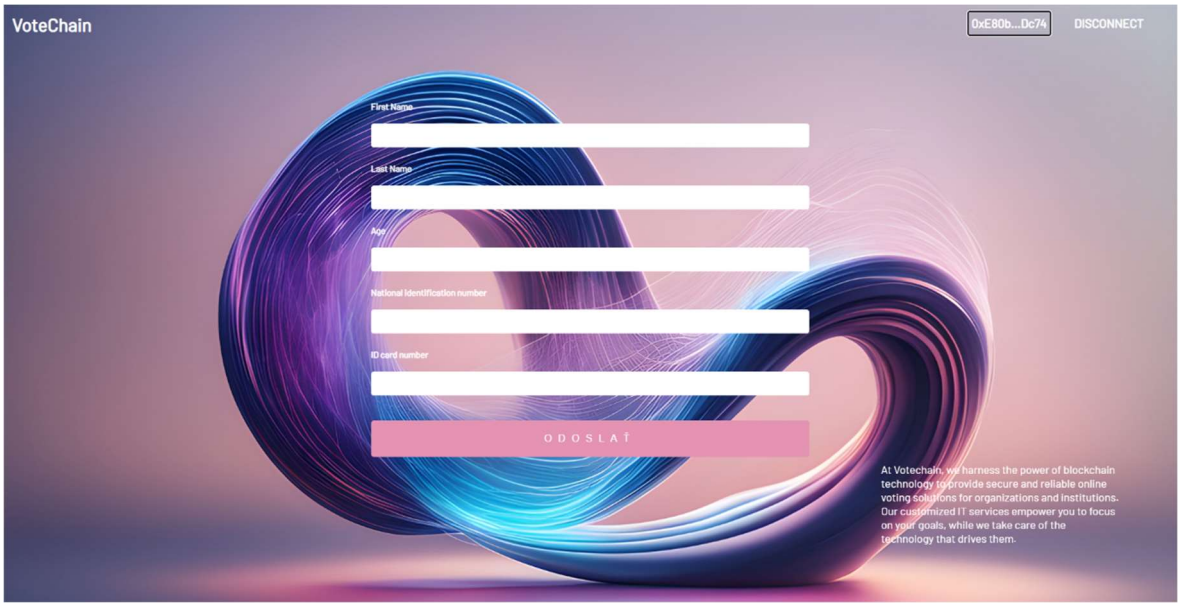
Ak už uplynula doba na hlasovanie, namiesto zoznamu kandidátov s možnosťou hlasovania sa zobrazia výsledky. Tieto výsledky sú zobrazené pomocou jednoduchého grafu. Užívateľ si taktiež môže skontrolovať tieto výsledky priamo na kontrakte volaním potrebnej metódy na zobrazenie kandidátov. Na zobrazenie výsledkov bola použitá React UI knižnica MUI X.



Obr. 22 Ukážka výsledku volieb

4.3.5 Overenie voliča

Po kliknutí na účet v pravom hornom rohu sa otvorí stránka s jednoduchým dotazníkom. Táto časť slúži na overenie voliča. Užívateľ musí vyplniť osobné informácie potrebné na jeho identifikáciu. Táto časť slúži iba ukážka systému, preto sa tieto informácie neposielajú na reálny overovací server, ale iba na vytvorenú službu, ktorá ho má napodobniť.



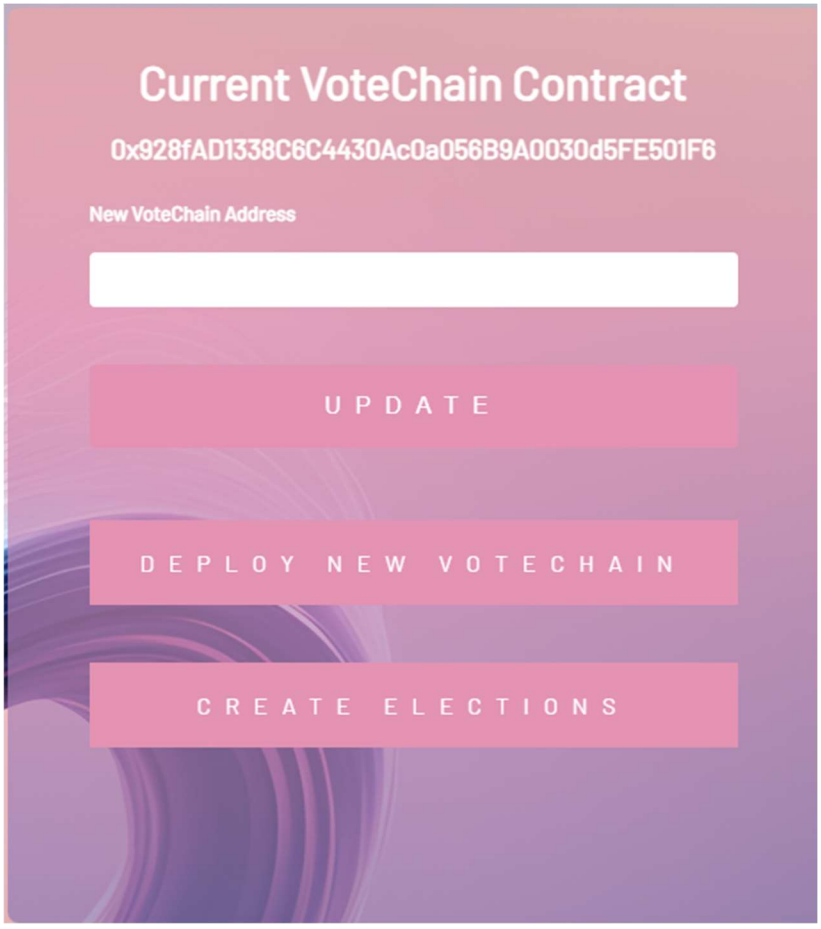
Obr. 23 Stránka na overenie voliča

Na tieto účely bol vytvorený testovací backend server, ktorý nezaznamenáva žiadne údaje. Tento server iba simuluje proces overenia dokladov. V reálnom nasadení by mohli byť žiadané ešte dodatočné informácie ako napríklad fotografia, scan občianskeho preukazu

alebo iného dokladu totožnosti. Po zadaní týchto informácii sa pošlú pomocou POST API end pointu na backendový server. Ten ich spracuje a upraví kontrakt podľa potreby. Následne odpovie na túto požiadavku naspať front end stránke. Tá výstup zobrazí pomocou kontextového toast okna a vráti ho naspať na úvodnú stránku s voľbami. Ak bol užívateľ úspešne overený, môže sa zúčastniť aj hlasovaní ktoré povoľujú hlasovanie iba overeným voličom.

4.3.6 Administrátorské rozhranie

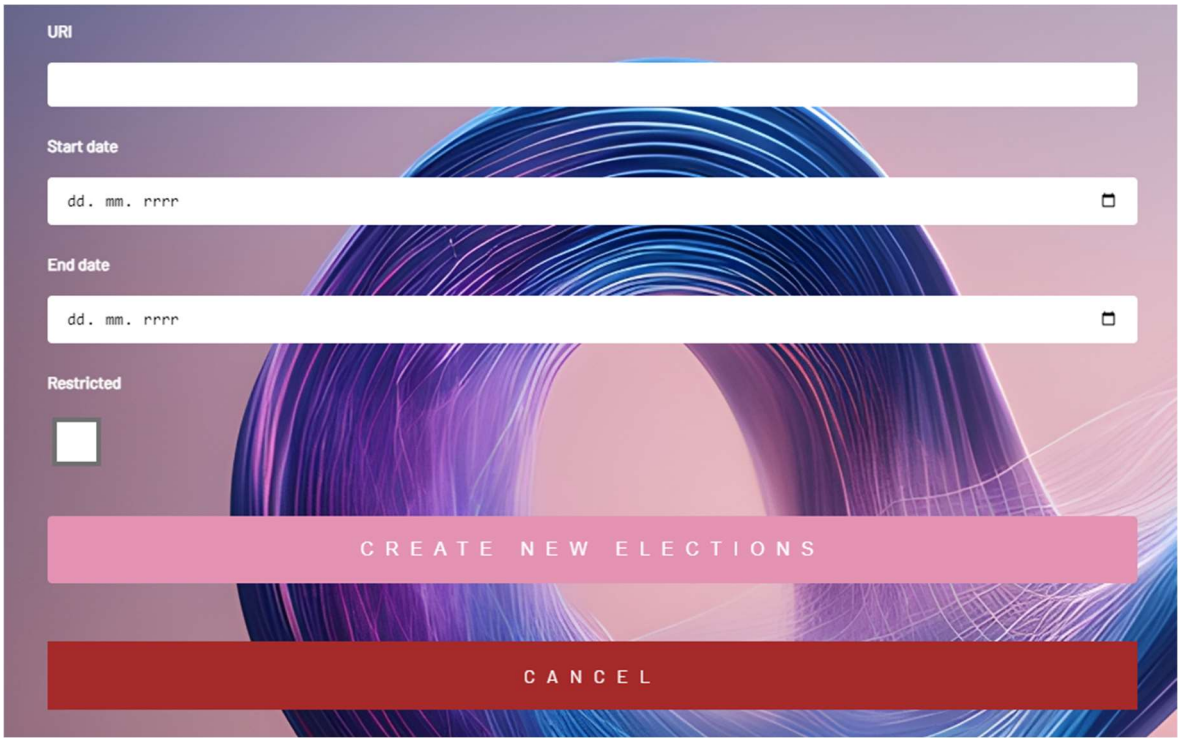
Ako posledné bolo implementované administrátorské rozhranie ktoré umožňuje správu kontraktov priamo vo webovej stránke online hlasovanie. Vďaka tomu používateľ nemusí hľadať externý nástroj ako napríklad Remix na vykonávanie niektorých akcií.



Obr. 24 Administrátorské rozhranie

Na webe je nastavený základný VoteChain použitý na testovacie účely. Ak sa ale používateľ rozhodne používať iný kontrakt, priamo v tomto rozhraní to môže vykonať. Na výber má buď zadať adresu už vytvoreného VoteChain smart kontraktu alebo môže vytvoriť svoj nový vlastný. Po nasadení sa užívateľ stáva vlastníkom a tento kontrakt sa nastaví ako práve používaný. Každý vlastník a administrátor má ďalej v správe pridávanie nových volieb

do aktuálneho VoteChainu. Po kliknutí na tlačidlo sa skryjú aktuálne dostupné voľby a zobrazí sa formulár na vytvorenie nového kontraktu volieb.



Obr. 25 Vytvorenie nových volieb

Administrátor musí vyplniť dané atribúty a následne potvrdiť dve transkacie. Prvá nasadzuje kontrakt, druhá slúži na pridanie novo vytvrených volieb do VoteChain správcu. Pri vytváraní sa všetky atribúty ako názov, obrázok a popis musia definovať v v json súbore uloženom na IPFS cloudovej službe. Do položky URI sa následne vkladá už iba hash súboru.

4.3.7 CI/CD

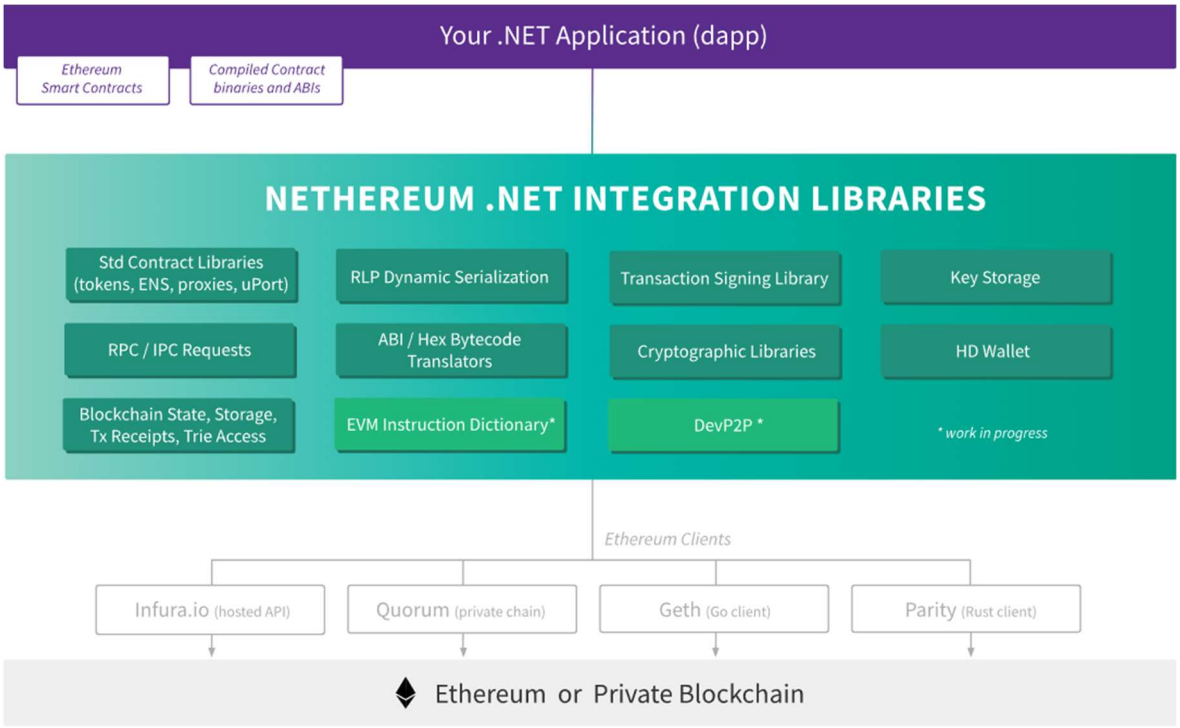
Na zrýchlenie vývoja a testovania bol použitý systém CI/CD. Ten umožnil automatické otestovanie a nasadenie novej verzie Front end webstránky. Tento systém bol vytvorený ako pipeline na Azure DevOps. Tento portál taktiež obsahuje aj ukladací priestor na repozitár v ktorom bol uložený kód. Toto nám uľahčilo vytvorenie nového "buildu" webovej stránky. Ten sa uložil do artefaktov. Po každom nahratí kódu sa vytvoril nový "build" a následne nahral na webový server. Tento webový server bol vytvorený ako služba na Azure portály. Následné bolo potrebné prepojiť tieto dve služby v "release" pipeline. Pred prvým spustením pipeline je ale nutné požiadať o túto službu ak ju chceme používať zadarmo. V tejto verzii nám povolí vykonávať maximálne jednu úlohu súčasne. Všetky ďalšie pipeline sa dajú do poradovníka.

4.4 Backend

Bol vytvorený backendový server, ktorý slúži ako mockup na overovanie identity používateľa. Bol implementovaný pomocou .NET API. Využíva knižnicu Nethereum, ktorá poskytuje potrebné nástroje a funkcie na interakciu s blockchainovou sieťou. Táto backend služba umožňuje simuláciu procesu overenia identity používateľa a poskytuje rozhranie, cez ktoré môže front end aplikácia zapisovať údaje na smart kontrakt. Je navrhnutý tak, aby bol flexibilný a jednoducho rozšíriteľný o ďalšie funkcionality podľa potrieb konkrétnych aplikácií. Aplikácia je rozdelená do rôznych programovacích častí pre lepšiu prehľadnosť. Obsahuje jediný kontrolér s POST rozhraním. Toto rozhranie príma číslo peňaženky užívateľa a JSON súbor v ktorom sa nachádzajú dáta o jeho identite. V reálne nasadenom prípade by bola táto služba nahradená potrebným buď štátnym, firemným alebo školským orgánom, ktorý by spravoval identifikáciu. Každý z nich by vyžadoval úplne iné informácie podľa potreby danej inštitúcie, ktorá by decentralizované online hlasovanie využívala.

4.4.1 Nethereum

Nethereum je open-source knižnica v jazyku C#, ktorá poskytuje nástroje a funkcie na interakciu s Ethereum blockchainom. Je navrhnutá tak, aby umožnila vývojárom jednoducho a efektívne komunikovať s Ethereum sieťou, vytvárať smart kontrakty, posielať transakcie alebo získavať údaje zo siete. Jeho fungovanie je založené na priamej komunikácii s Ethereum uzlami pomocou Ethereum RPC (Remote Procedure Call) protokolu. Správa smart kontraktov, blockchainu a Etherea pomocou C# a .NET nie je bežnou praktikou. Kvôli tomu je zvyčajne ťažké nájsť podobný problém alebo rovno riešenie na tento problém. Na podrobné fungovanie a príklady použitia je vhodným zdrojom oficiálna dokumentácia Netherea. Tá ale tiež nie je plne dostačujúca. Ďalším zdrojom informácii je tvorca tejto knižnice, ktorý je aktívny na fórach a pomáha pri riešení rôznych chýb. Na bežné fungovanie je ale táto knižnica dostačujúca. Taktiež je možné vytvoriť aj zložitejšie aplikácie. Na tieto je ale potrebné väčšie množstvo skúseností a času na implementáciu.



Obr. 26 Nethereum

V backendovej aplikácii sú zatiaľ využívané iba dve hlavné funkcionality. Najskôr je potrebné overiť, či už užívateľ náhodou nebol overený, aby sa zbytočne neplytvalo poplatkami za gas. Ak užívateľ ešte nebol overený, tak prebehne druhá časť kódu ktorá pošle transakciu na overenie voliča. V tejto transakcii sa posiela iba číslo Ethereum účtu pri ktorom sa zmení booleovská hodnota v smart kontrakte. Vďaka tomu nie je možné žiadnym spôsobom prepojiť dáta o užívateľovi k jeho číslu peňaženky, keďže sa nikde nezaznamenávajú jeho osobné informácie. Toto riešenie zabezpečuje pseudo anonymitu hlasu.

Ako funguje Nethereum

Najčastejšou funkciu, ktorá sa vyžaduje od knižnice na správu blockchainu a web3 aplikácií je posielanie transakcií alebo čítanie stavu a atribútov nejakého smart kontraktu. Keďže posielanie transakcií vyžaduje poplatok za gas a vlastník back end služby nemôže podpisovať každú transakciu manuálne, potrebuje vytvoriť v kóde Ethereum účet ktorý to bude vykonávať za neho automatizovane. Na jeho správne nastavenie Nethereum knižnica vyžaduje zadať privátny kľúč ktorý slúži práve na tieto podpisy. Taktiež by mal účet obsahovať dostatočné množstvo Ethera alebo ošetrenie chýb v prípade jeho nedostatku.

Tento Nethereum účet slúži na vytvorenie pripojenia na blockchain pomocou Web3 objektu. Ďalším potrebným atribútom je url adresa providera/poskytovateľa.

Provider je abstrakcia pripojenia k sieti Ethereum, poskytujúca stručné a konzistentné rozhranie pre štandardnú funkčnosť uzla Ethereum. Momentálne medzi najznámejších a najlepších poskytovateľov patrí Alchemy, Etherscan, Infura, Pocket alebo Quorum. Každý z nich vyžaduje iné parametre. Vytvára sa FallbackProvider, ktorý je pripojený k čo najväčšiemu počtu backendových služieb. Keď sa požiadavka odosiela, súčasne sa posiela na viacero backendov. Keď prichádzajú od každého back endu odpovede, kontroluje sa, či súhlasia. Ak bolo dosiahnuté kvórum, to znamená, že dostatočné množstvo backendov súhlasí, odpoveď je poskytnutá aplikácii. [22]

Tab. 2 Poskytovatelia

Poskytovateľ	**Parametre**
Alchemy	API Token
Etherscan	API Token
Infura	Id projektu + heslo
Pocket	Pocket Network Application ID
Quorum	Počet back endov, ktoré musia súhlasiť

Na naše účely bola použitá služba Alchemy, kde bolo najskôr nutné vytvoriť účet. Iba takto bolo možné získať potrebný API Token. Pomocou tohto poskytovateľa sa pripájame na Ethereum API. Toto API zodpovedá JSON-RPC štandardu. Vďaka použitiu Nethereum knižnice ale netreba používať priame pripojenie s RPC(remote procedure call) volaniami.

Podobne ako pri používaní Wagmi na front ende, aj tu je potrebné mať definované ABI. Pomocou neho sa vytvorí objekt kontraktu. Pre volanie konkrétnej funkcie definovanej v ABI smart kontraktu, vytvorí sa jej inštancia pomocou: konktrakt.GetFunction("názov funkcie"). Pri volaní pure alebo view metódy, ktorá nevyžaduje poplatok za gas a iba číta informácie zo smart kontraktu, stačí iba asynchrónne zavolať túto funkciu. Pri transakciách ktoré vyžadujú gas na fungovanie, je predtým ešte nutné odhadnúť cenu. Tieto funkcie vracajú transakčný hash, ktorý je možné skontrolovať pomocou blockchain explorerov.

```
public UserService()
{
    account = new Account(privateKey);
    web3 = new Web3(account,url);
    const string fileName = "VoteChain.abi";
    string path = Path.GetFullPath(fileName);
    _abi = System.IO.File.ReadAllText(path);
    contract = web3.Eth.GetContract(_abi, contractAddress);
}

2 references
public async Task<IActionResult> AuthorizeUser(string wallet, UserAuthDTO user)
{
    //CHECK FOR USER INFORMATION
    //
    //
    //
    //
    //
    //
    ////////////////////////
    var isAuthorizedFunction = contract.GetFunction("isAuthorized");
    var isAuthorized = await isAuthorizedFunction.CallAsync<bool>(wallet);
    if (isAuthorized)
    {
        return Ok("User "+wallet+ " is already authorized");
    }
    var function = contract.GetFunction("authorizeVoter");
    var gas = await function.EstimateGasAsync(senderAddress, null, null, wallet);
    var transactionHash = await function.SendTransactionAndWaitForReceiptAsync(senderAddress, gas, null, null, wallet);
    return Ok("User " + wallet + " authorized");
}
```

Obr. 27 Použitie Nethereum knižnice na posielanie transakcií

4.4.2 Nasadenie

Podobne ako webová stránka, tak aj backend server bol nasadený na bezplatnú verziu MS Azure servera. Keďže sa kód tejto služby nebude často meniť a aj vývoj trval krátko, nebolo potrebné implementovať kontinuálny deployment. Nasadenie prebieha manuálne priamo v editore Visual Studia 2022. Po prvom nasadení kód nefungoval správne. Aplikácia nedokázala načítať ABI súbor. Pri buildovaní projektu sa totižto tento súbor nevytvoril. Bolo potrebné explicitne nastaviť aby sa súbor s ABI kopíroval do build priečinku. Toto riešenie opravilo chybu a backend následne fungoval správne.

4.4.3 CORS

Backendový server sa nachádza na inej doméne ako webová stránka. Všetky moderné prehliadače kvôli bezpečnosti aplikujú CORS mechanizmus. Ten blokuje všetky API požiadavky na server ktorý sa nachádza na inej doméne. Tento mechanizmus umožňuje webovým stránkam vyžiadať si povolenie na prístup k určitým zdrojom na serveri, pričom server môže poskytnúť alebo odmietnuť túto žiadosť na základe definovaných pravidiel. CORS funguje prostredníctvom HTTP hlavičiek, ktoré sú zaslané medzi klientom a serverom. Kvôli tomu sa musel povoliť prístup na backend servery konkrétnej domény na ktorom sa nachádza front end. Microsoft Azure služba na to má rýchle a jednoduché rozhranie v ktorom je možne definovať povolené url adresy.

4.5 Výsledky

4.5.1 Skúsenosti

Práca so smart kontraktmi bola najjednoduchšou časťou tohto projektu. Ich tvorba je už dlhodobo ustálená a neprechádzajú zásadnými zmenami. Na ich tvorbu je aj množstvo rôznych nástrojov, ktoré sú dobre zdokumentované. Taktiež sú dostupné zaujímavé spôsoby ako sa dá smart kontrakty písať. Jedným z nich je aj webová stránka CryptoZombies. Nachádzajú sa tu tutoriály ako začať pracovať s vývojom smart kontraktov. Ďalej prechádzajú k oveľa pokročilejším témam. Snažia sa pretvoriť učenie zábavným a odmeňujúcim spôsobom. Za každú lekciu užívateľ dostane odmenu v podobe NFT.

Toto isté sa už bohužiaľ nedá povedať o budovaní front end a back end aplikácii, ktoré spolupracujú s blockchainom. Len v priebehu posledného roka sa front end knižnice úplne zmenili minimálne 4 krát. Vznikajú nové požadované funkcionality od vývojárov ktoré nestíhajú byť dobre dokumentované. Staré funkciu sú buď úplne zmenené alebo stratia podporu a nefungujú vôbec. Z tohto dôvodu treba byť pripravený, že počas vývoja alebo v budúcnosti sa bude musieť základná logika projektu prerábať. Aj v tomto projekte sa začalo pracovať s knižnicou wagmi, ktorá niekedy obsahovala všetky potrebné funkcionality. V poslednej verzii ale odstránili nasadzovanie kontraktov. Preto bolo nutné použiť novú knižnicu Viem, ktorej ale chýbali iné funkcie. Kvôli tomu sa muselo zabezpečiť aby tieto dve knižnice vedeli spolupracovať. Táto úloha nie vždy je jednoduchá a prehľadná. V kóde sa potom nachádza veľké množstvo zle čitateľných metód. Taktiež ak vývojár už vytvoril nejaký projekt v minulosti, informácie z neho nemôže použiť na ďalší. Stále sa musí učiť ako pracovať s najnovšími verziami knižníc.

4.5.2 Porovnanie ceny volieb

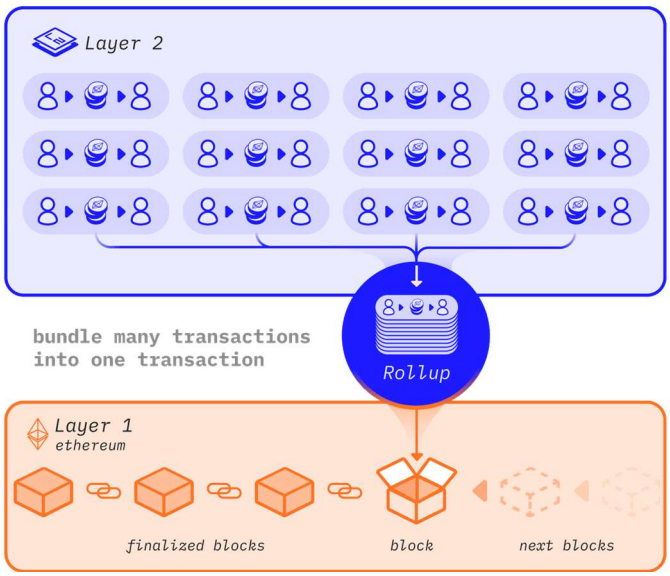
Podľa harmonogramu organizačno-technického zabezpečenia volieb prezidenta SR v roku 2024 schváleného vládou SR budú náklady na organizáciu volieb 18,6 milióna eur. Z rozpočtu ministerstva vnútra SR pôjde na voľby 16 195 505 eur a z rozpočtu Štatistického úradu SR 2 421 000 eur [23]. Pri volebnej účasti 2 265 656 v prvom kole a 2 671 279 voličov v druhom kole je možné vypočítať cenu jedného hlasu. Náklady na jeden hlas týchto volieb dosiahli hodnotu 3.77 €. Pri testovaní blockchain hlasovania náklady na odoslanie jedného hlasu spotrebovali približne 76000 gasu. Pri aktuálnej cene Etherea 3220 € táto jedna transakcia stojí približne 6,36 €. Z tohto vyplýva, že pri momentálnej vysokej cene Etherea sa finančne viac oplatí organizovať klasické papierové voľby. Pri poklese ceny Etherea sa

ale tieto náklady vyrovnávajú. Riešením tohto problému by mohli byť L2 vrstvy Ethereum siete. Poplatky na tejto druhej vrstve sú násobne nižšie. Pri rovnakom množstve spotrebovaného gasu by jeden hlas stál iba 0.0245 € ak by bola použitá sieť Arbitrum One. Kompromisom tohto riešenia je ale znížená bezpečnosť hlasovania.

4.5.3 L2 vrstva

L2 (Layer 2) vrstvy predstavujú druhý stupeň škálovateľnosti nad samotnou Ethereum blockchain sieťou. Tieto vrstvy sú navrhnuté s cieľom riešiť obmedzenia výkonu, vysoké poplatky za transakcie a oneskorenie potvrdení transakcií, ktoré môžu byť sprevádzané na hlavnej vrstve Ethereum. Hlavná sieť (L1) je schopná spracovať iba približne 15 transakcií za sekundu. Keď je dopyt po používaní Etherea vysoký, sieť sa preťaží, čo vedie k zvýšeniu transakčných poplatkov a tým znemožňuje používanie užívateľom, ktorí si tieto poplatky nemôžu dovoliť. Vrstvy 2 sú riešenia, ktoré znížia tieto poplatky spracovaním transakcií mimo blockchainu vrstvy 1. [24]

Rollups



Obr. 28 L2 Rollups [24]

Rollup zoskupuje stovky transakcií do jedinej transakcie L1. Tým sa transakčné poplatky L1 rozdelia medzi všetkých účastníkov transakcie, čo v konečnom dôsledku znižuje náklady pre každého používateľa. Transakcie v rollupoch sa uskutočňujú mimo L1, ale údaje týchto transakcií sú odoslané do vrstvy 1. Tým, že sa údaje o transakciách nahrávajú do L1, majú rollupy bezpečnosť rovnakú ako Ethereum. Je to preto, ak sú údaje nahraté do L1, vrátenie transakcie v rollupe L2 vyžaduje celé vrátenie transakcie z L1. [24]

Záver

Výsledkom tejto diplomovej práce je plne funkčná aplikácia na online hlasovanie. Aplikácia využíva plne decentralizovanú blockchain technológiu s využitím Ethereum smart kontraktov. Je zameraná na zabezpečenie základných požiadaviek hlasovania. Jej implementácia umožňuje vytváranie volieb, pridávanie kandidátov a hlasovanie za týchto kandidátov. Aplikáciu je možné taktiež obsluhovať pomocou webovej stránky vytvorenej v Reacte.

Táto diplomová práca poukazuje aj na význam decentralizovanej webovej aplikácie pre online hlasovanie v modernizácii volebných procesov. S nástupom blockchain technológie sa otvárajú nové možnosti v oblasti elektronického hlasovania, ktoré prinášajú transparentnosť, nezvratnosť a bezpečnosť.

Významným cieľom práce nebolo iba navrhnúť a vytvoriť decentralizovanú aplikáciu pre online hlasovanie, ale aj poukázať na potenciál blockchain technológii. Vývoj v oblasti blockchainu a jeho využitie nie je obmedzené len na finančné transakcie, ale otvára dvere aj pre iné oblasti, ako je elektronické hlasovanie. Vďaka blockchainu je možné zabezpečiť dôveryhodnosť, bezpečnosť a transparentnosť všetkých procesov ktoré sa na sieti vykonávajú. Vytvorenie nezvratných záznamov zabezpečuje spravodlivé započítanie každého hlasu a možnosť overenia autenticity všetkých údajov.

Taktiež sa práca snaží poukázať na nový typ decentralizovaných aplikácií. Decentralizované aplikácie bežiace na blockchain sieti majú množstvo výhod oproti centrálne riadeným systémom, vrátane odolnosti voči cenzúre a výpadkom. Tieto aplikácie majú ešte dlhú cestu pred sebou. Zatiaľ sú používane len minimálne ale ich dosah pre bežnú verejnosť sa každý deň zvyšuje.

Aby aplikácia mohla naozaj ostať decentralizovaná a transparentná, musí túto vlastnosť z časti vymeniť za zníženie anonymity. Plne anonymný bezpečný a zároveň transparentný systém nie je praktický možné vytvoriť. Na príkladoch použitia z celého sveta bolo možné vidieť tento kompromis. Ak by sme chceli dosiahnuť úplnú anonymitu hlasu, bolo by potrebné každý hlas šifrovať. Keďže môžeme prezerať celú históriu blockchainu, ak by hlasy boli nezašifrované, vedeli by sme zistiť ktorá peňaženka hlasovala za ktorého kandidáta. Pri vynaložení dostatočných prostriedkov je taktiež možné spojiť si číslo tejto peňaženky s reálnou osobou. V tomto prípade by voľby úplne stratili status toho že sú tajné.

Už z definície deterministickosti a izolovanosti blockchainu vychádza, že je nemožné tieto hlasy bezpečne šifrovať a dešifrovať priamo v smart kontrakte. Z tohto vychádza jediné riešenie a to je centrálny server ktorý by spravoval verejne a privátne kľúče. Následne by dešifroval hlasy po skončení volieb. Toto riešenie ale kontradikuje s decentralizovanou vlastnosťou aplikácie.

5 Zoznam použitej literatúry

[1] „Council of Europe,“ 14 Jún 2017. [Online]. Available: https://www.coe.int/en/web/electoral-assistance/e-voting.

[2] „Valimised,“ [Online]. Available: https://www.valimised.ee/en/internet-voting/more-about-i-voting/introduction-i-voting.

[3] Confederaziun Svizra, [Online]. Available: https://www.ch.ch/en/votes-and-elections/e-voting/#further-information.

[4] N. Goodman, „centreforedemocracy,“ The Centre for e-Democracy & The University of Toronto, August 2016. [Online]. Available: http://www.centreforedemocracy.com/wp-content/uploads/2016/08/IVP_Report.pdf.

[5] M. Baxter, TVO’s southwestern Ontario Hubs reporter, 4 október 2018. [Online]. Available: https://www.tvo.org/article/how-e-voting-is-taking-over-ontario-municipal-elections.

[6] C. Butler, „Ontario civic elections: the problem with online voting,“ CBC News, 4 Apríl 2018. [Online]. Available: https://www.cbc.ca/news/canada/london/london-ontario-online-voting-1.4598787.

[7] S. Nakamoto, „Bitcoin: A Peer-to-Peer Electronic Cash System,“ Október 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[8] „How does Bitcoin work?,“ bitcoin.org, [Online]. Available: https://bitcoin.org/en/how-it-works.

[9] V. Buterin, „Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform,“ 2014. [Online]. Available: https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf.

[10] @nhsz, @minimalsm a @atharvadeosthale, „INTRODUCTION TO DAPPS,“ 15 August 2023. [Online]. Available: https://ethereum.org/en/developers/docs/dapps/.

[11] @nhsz, „GAS AND FEES,“ 19 Marec 2024. [Online]. Available: https://ethereum.org/en/developers/docs/gas/.

[12] The Solidity Authors, „Solidity,“ 2023. [Online]. Available: https://docs.soliditylang.org/en/v0.8.25/index.html.

[13] Protocol Labs, „What is IPFS,“ 2023. [Online]. Available: https://docs.ipfs.tech/concepts/what-is-ipfs/#defining-ipfs.

[14] Protocol Labs, „IPFS and the problems it solves,“ 2023. [Online]. Available: https://docs.ipfs.tech/concepts/ipfs-solves/#ipfs-and-the-problems-it-solves.

[15] S. Simkins, „Welcome to Pinata!,“ 2024. [Online]. Available: https://docs.pinata.cloud/introduction.

[16] D. Herbert, „What is React.js? Uses, Examples, & More,“ 13 November 2023. [Online]. Available: https://blog.hubspot.com/website/react-js.

[17] Wagmi team, „Why Wagmi,“ 24 Marec 2024. [Online]. Available: https://wagmi.sh/react/why.

[18] S. J. Bigelow, „Microsoft Azure,“ Október 2022. [Online]. Available: https://www.techtarget.com/searchcloudcomputing/definition/Windows-Azure.

[19] Microsoft team, „What is Azure DevOps?,“ 1 Február 2024. [Online]. Available: https://learn.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops?view=azure-devops.

[20] Microsoft team, „What is Azure Pipelines?,“ 1 Február 2024. [Online]. Available: https://learn.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops.

[21] J. Sahu, „Creating React App using Vite and PNPM,“ 2 September 2023. [Online]. Available: https://medium.com/@sahu.jyotirmaya26/creating-react-app-using-vite-and-pnpm-746bb0f9a0c2.

[22] R. Moore, „Providers,“ 6 Apríl 2023. [Online]. Available: https://docs.ethers.org/v5/api/providers/.

[23] TASR, SITA, „Prezidentské voľby budú stáť vyše 18 miliónov eur,“ 18 Január 2024. [Online]. Available: https://domov.sme.sk/c/23270235/prezidentske-volby-budu-stat-vyse-18-milionov-eur.html.

[24] ethereum.org, „Ethereum pro každého,“ 26 Marec 2024. [Online]. Available: https://ethereum.org/cs/layer-2/.

Zoznam príloh

Prílohy

Príloha A: Obsah DVD

Priložené DVD obsahuje:

- Práca v elektronickej podobe (formát PDF)
- Kód kontraktov (.sol)
- Kód Front End (React)
- Kód Back End (ASP .NET)